WARWICK
THE UNIVERSITY OF WARWICK

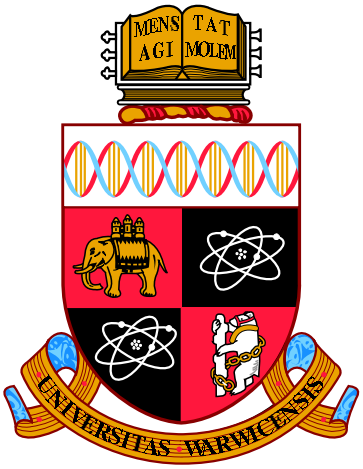
A Thesis Submitted for the Degree of PhD at the University of Warwick

Permanent WRAP URL:
http://wrap.warwick.ac.uk/128793

Copyright and reuse:
This thesis is made available online and is protected by original copyright.
Please scroll down to view the document itself.
Please refer to the repository record for this item for information to help you to cite it.
Our policy information is available from the repository home page.

For more information, please contact the WRAP Team at: wrap@warwick.ac.uk

warwick.ac.uk/lib-publications



Probabilistic modelling of uncertainty with Bayesian nonparametric machine learning

by

Charles Gadd

Thesis

Submitted to the University of Warwick

for the degree of

Doctor of Philosophy

School of Engineering

December 2018

THE UNIVERSITY OF WARWICK

Acknowledgments

Foremost I would like to thank my supervisors, Dr Sara Wade and Dr Akeel Shah, for their guidance, support, and stimulating discussions. I am also grateful to all my colleagues, both within and outside the department, who have been involved with this work.

I gratefully acknowledge the financial support from the EPSRC funding body that has made completion of this doctorate possible. Additionally, data collection and sharing for the Alzheimer's disease study was funded by the Alzheimer's Disease Neuroimaging Initiative (ADNI) (National Institutes of Health Grant U01 AG024904). I acknowledge the funding contributions of ADNI supporters (`adni-info.org/Scientists/ADNISponsors.aspx`).

Abstract

This thesis addresses the use of probabilistic predictive modelling and machine learning for quantifying uncertainties. Predictive modelling makes inferences of a process from observations obtained using computational modelling, simulation, or experimentation. This is often achieved using statistical machine learning models which predict the outcome as a function of variable predictors and given process observations. Towards this end Bayesian nonparametric regression is used, which is a highly flexible and probabilistic type of statistical model and provides a natural framework in which uncertainties can be included.

The contributions of this thesis are threefold. Firstly, a novel approach to quantify parametric uncertainty in the Gaussian process latent variable model is presented, which is shown to improve predictive performance when compared with the commonly used variational expectation maximisation approach. Secondly, an emulator using manifold learning (local tangent space alignment) is developed for the purpose of dealing with problems where outputs lie in a high dimensional manifold. Using this, a framework is proposed to solve the forward problem for uncertainty quantification and applied to two fluid dynamics simulations. Finally, an enriched clustering model for generalised mixtures of Gaussian process experts is presented, which improves clustering, scaling with the number of covariates, and prediction when compared with what is known as the alternative model. This is then applied to a study of Alzheimer's disease, with the aim of improving prediction of disease progression.

Abbreviations

- AD ..... Alzheimer's Disease
- ADNI ..... Alzheimer's Disease Neuroimaging Initiative
- ANN ..... Artificial neural network
- ARD ..... Automatic relevance determination
- BNP ..... Bayesian nonparametric
- CN ..... Cognitively normal
- DP ..... Dirichlet process
- EDP ..... Enriched Dirichlet process
- EMCI ..... Early mild cognitive impairment
- ESS ..... Elliptical slice sampling
- FD ..... Finite difference
- GLM ..... Generalised linear model
- GP ..... Gaussian process
- GPLVM ..... Gaussian process latent variable model
- HMC ..... Hamiltonian Monte Carlo
- KDE ..... Kernel density estimation
- KL ..... Kullback-Leibler
- KLE ..... Karhunen-Loève expansion
- LMCI ..... Late mild cognitive impairment
- LTSA ..... Local tangent space alignment

- MAP ..... Maximum a posteriori
- MC ..... Monte Carlo
- MCMC ..... Markov Chain Monte Carlo
- ML ..... Maximum likelihood
- MML ..... Maximum marginal likelihood
- MMSE ..... Mini-mental state exam
- PDE ..... Partial differential equation
- PM ..... Pseudo-marginal
- PMMC ..... Pseudo-marginal Monte Carlo
- PPCA ..... Probabilistic principal component analysis
- sGPLVM ..... Supervised Gaussian process latent variable model
- UQ ..... Uncertainty quantification
- VEM ..... Variational expectation maximisation
- VI ..... Variational inference

Notation

Within this section the nomenclature and notation commonly found throughout this thesis is presented. Where interchangeable the letter 'a' is used. Notation not outlined here is defined upon use.

Roman symbols

$\mathbf{a}$:	Vector
$\mathbf{A}$:	Matrix
a:	Scalar
$\mathcal{A}$:	Vector space in which vector $\mathbf{a}$ lies
k_a:	For some vector $\mathbf{a} \in \mathcal{A}$, then k_a is the dimension of the vector space
$T_{\mathbf{a}}\mathcal{A}$:	The tangent space (a linear subspace) of vector space $\mathcal{A}$ at $\mathbf{a}$
$\mathbf{X}$:	Dataset inputs
$\mathbf{Y}$:	Dataset outputs
$\mathcal{D}$:	Dataset $\{X, Y\}$
N:	Number of samples in the training dataset

Greek symbols

θ, σ, β:	The Gaussian process hyperparameters
$\mathbf{\Theta}$:	The joint set of Gaussian process hyperparameters
α:	The Dirichlet process mass/concentration parameter

Superscripts

$a^{(i)}$:	ith value in a series (e.g. Markov Chain, importance samples, etc.)
a^{ML}:	The maximum marginal likelihood estimate of hyperparameter a

Subscripts

$\mathbf{a}_i$:	The ith element of a vector
$\mathbf{A}_i$:	The ith row of a matrix (e.g. $\mathbf{X}_i$ is the ith sample's covariates)
$\mathbf{A}_{:,j}$:	The jth column of a matrix (e.g. $\mathbf{X}_{:,j}$ is the jth covariate of all samples)
$\mathbf{A}_{i,j}$:	The element of a matrix on the ith row and jth column

Other symbols

N:	The Gaussian distribution
Dir:	The Dirichlet distribution
Bern:	The Bernoulli distribution
Ga:	The Gamma distribution
Beta:	The Beta distribution
GP:	The Gaussian process
DP:	The Dirichlet process
EDP:	The enriched Dirichlet process
$\mathbb{R}$:	The real numbers
$\mathcal{O}$:	Big-O notation

Functions

$K(\cdot,\cdot)$:	A kernel function
$\mathbf{K_a}$:	A kernel function evaluated at points in $\mathcal{A}$.
$\mathbf{K_{a*}}$:	A kernel function evaluated between training and test points in $\mathcal{A}$.
$\mathbf{K_{i,j}}$:	A kernel function evaluated at indexing points (or subsets) i and j

Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

1.1 Motivation

Machine learning is the act of using algorithms and models which allow computers to 'learn' based on a set of experiences, where experiences often exist in the form of data. In this context, learning is the process of gaining an understanding of a task through the building of a computational model of training data. This learnt model can then be used to make predictions or decisions without requiring rules to make them being specifically programmed, as would be the case in a rule-based system. The work presented in this thesis lies in the domain of statistical machine learning, in which machine learning methods are combined with statistical techniques under the assumption of statistical regularity in the data. Specifically, the work presented here lies at the intersection of Bayesian nonparametrics (BNP) and machine learning. The upshot of BNP machine learning is a natural probabilistic framework for an interpretable inclusion of uncertainties though probability theory, with the framework simplicity of Bayesian models, while being able to model the complexity of real world phenomena using nonparametrics. This approach can be used to infer unknown quantities, adapt models, learn from the data and make predictions.

Models are not perfect and uncertainty can manifest in many areas of the modelling process. A data-driven model serves as an approximation to a system and this comes with a number of uncertainties. Often these uncertainties are grouped into two disjoint groups, *epistemic uncertainty* and *aleatoric uncertainty*. Epistemic uncertainty (from the Greek word for "knowledge" and sometimes referred to as reducible or systematic uncertainty) is uncertainty which can be explained with more knowledge. Aleatoric uncertainty (from the Latin word for "dice" and sometimes referred to as irreducible or statistical uncertainty) is uncertainty due to unknowns

that change each time the experiment is run. For example, an infinite number of dice rolls does not remove the stochastic nature of rolling a dice. Another common example is an arrow's impact point which, given initial firing parameters, will vary due to seemingly random vibrations in the arrow shaft. In this case the uncertainty occurs due to the lack of knowledge. However, once this information can be obtained it becomes an epistemic uncertainty. Prevalent examples of uncertainty encountered when modelling include:

- Data noise. For example, from measurement imprecision, human error, or missing explanatory variables in the data set.
- Out of distribution/interpolation uncertainty. For example, in data-driven models predictions away from the training samples should have a higher predictive uncertainty to reflect the reduced available information.
- Model structure/distributional uncertainty. For example, a model may assume heterogeneous noise, distributional form, regularity, stationarity, and a function's smoothness or form. Additionally there may be approximations to a model which introduces further uncertainties.
- Model parameter uncertainty. For example, there may be a large number of parameters (and therefore models) which can explain the observed data. Similarly, in the nonparametric setting with an infinite dimensional parameter space, there is uncertainty associated with the choice of hyperparameters.

A principled approach to understanding, quantifying, reducing and modelling these uncertainties is critical for many scenarios. Obvious examples include any high risk decision making task where it is crucial that a model output can be trusted, such as in performing a medical diagnosis or assisted driving. These themes are core to this thesis and a deeper understanding allows us to answer many questions, such as whether a model can be trusted, if predictions are uncertain, or if approximations to a model are accurate.

This work is motivated from both a methodological and practical context centred around probabilistic modelling of uncertainty. Multiple tools are developed which shed light on modelling uncertainty and these tools are applied to problems including fluid dynamic emulation and an Alzheimer's Disease Neuroimaging Initiative (ADNI) challenge to predict the decline in cognitive impairment.

1.2 Thesis structure

The first part of this thesis (chapter 2) introduces the fundamentals upon which this work builds. This includes a brief introduction to BNP modelling, two popular prior processes commonly used in BNP models and a brief introduction to some schemes used to perform posterior inference in these models.

Chapter 3 introduces a novel Bayesian framework for inference with a supervised version of the Gaussian process latent variable model (GPLVM). This is motivated by weaknesses in the use of: point estimates to hyperparameters; approximations of the estimates, often through non-convex optimisation; and model approximations, through variational expectation maximisation. GPLVM is a hierarchical model in which hyperparameters and latent variables are heavily correlated. The proposed framework overcomes these correlations using a collapsed Metropolis-within-Gibbs sampler, with an unbiased pseudo estimate for the marginal likelihood that approximately integrates over the latent variables and samples the hyperparameter posterior. Conditional on these samples, the framework continues by uncollapsing the model with elliptical slice sampling (ESS) to explore the posterior of the latent variables. The procedure is demonstrated on simulated examples, showing the ability to capture uncertainty and multimodality of the hyperparameters. Additionally, the approach improves the accuracy of predictions when compared with the state-of-the-art inference techniques for GPLVM, which often come with the aforementioned weaknesses.

Following this, chapter 4 develops a surrogate modelling approach to regression in high dimensional output spaces which lie on a manifold. This is then used to construct a framework to solve the forward problem of uncertainty quantification (UQ), in which input uncertainty is propagated through a model to predict the uncertainty in the system response. The approach obtains a lower dimensional latent representation of sample outputs using a feature extraction step using local tangent space alignment (LTSA), a nonparametric (but non-Bayesian) approach to manifold learning. These extracted features can then be used in conjunction with BNP Gaussian process (GP) emulation. This is then demonstrated on groundwater flow models involving a stochastic input field (e.g. the hydraulic conductivity) to capture the output field (e.g. the pressure head). A Karhunen-Loève expansion (KLE) for a log-normally distributed input field is used. Two examples are presented to demonstrate the accuracy: a Darcy flow model with contaminant transport in 2 spatial dimensions and a Richards equation model in 3 spatial dimensions.

Finally, chapter 5 presents an infinite mixture of GP experts model, which

partitions the input space into regions where stationary and heterogeneous noise assumptions of the GP must only hold in each region. What is known as an alternative model is presented, where the joint distribution of the inputs and targets is modelled explicitly. Whilst this modelling choice gives the ability to handle missing data and answer inverse problems, the local input model causes 1) the model to scale poorly with increasing input dimension and 2) the creation of an unnecessary number of experts, degrading the predictive performance and increasing uncertainty. To address the former, local independence assumptions of the inputs are made. This also allows for the inclusion of multiple input types. For the latter, the enriched Dirichlet process is utilised, allowing for a nested partitioning scheme and an analytically computable allocation rule. This allows the development of efficient sampling algorithms for posterior inference. These advantages are demonstrated on a highly non-linear toy example with increasing input dimension and an Alzheimer's challenge to predict decline in cognitive impairment.

1.3 Associated publications and software

- The work presented in chapter 3 is based on Gadd et al. [2018] (in preparation). This paper presents a scheme for full Bayesian inference for the supervised GPLVM, which can be generalised to other models, such as the deep Gaussian process.
- The work presented in chapter 4 is based on Gadd et al. [2018]. This paper uses GP emulation for UQ tasks in a highly non-linear ground water flow problem, where the output space is high dimensional.
- The work presented in chapter 5 is based on work in preparation, where an enriched mixture model of generalised GP experts is presented.
- Software to implement the models proposed in this thesis is made freely available on an open source license at gitlab.com/charles1992.

Chapter 2

Pre-requisites

This chapter serves as a gentle introduction to the topics this thesis builds upon. The following sections introduce Bayesian nonparametric models; review two of the most popular priors (the Gaussian and Dirichlet process) and introduce some methods used to perform Bayesian inference in this family of models.

2.1 Bayesian analysis

Bayesian analysis is a self-contained paradigm for statistics that approaches unknown parameters probabilistically, treating them as random variables and examines properties of the unknown random parameters conditioned on a set of observed data samples. Conversely, classical (frequentist) approaches treat parameters as unknown but fixed values and aim to find estimators of the fixed parameters with desirable properties that average over all potential data samples.

Arguments for using the Bayesian approach are extensive and compelling. Firstly, Bayesian analysis allows for a natural quantification of uncertainty, giving a direct statement of the believed hypothesis probability. Classical approaches instead provide significance levels (p-values), or the probability of a Type I or Type II error, which can *sometimes* then be *indirectly* related to the hypothesis probability. For example, when the hypothesis is on an unknown population parameter Bayesian methods provide an interval estimation (known as the credible interval) of plausible values which can be interpreted subjectively by the practitioner as the probability that, given the data, the true value lies in the credible interval. In classical approaches the interval estimation (confidence interval) is an estimate of the population parameter, but does not necessarily include the true value. If repeated for different data samples, the fraction that contains the true value will tend towards

the confidence level, which is subjectively chosen by the practitioner. This relies on an asymptotic approximation, but the Bayesian approach provides inferences that are conditional on the data and are exact.

In almost all statistical problems, not using prior information can lead to obtaining weak or nonsensical results. The Bayesian paradigm allows for a natural, well-defined and more interpretable inclusion of prior information, which requires an appropriate prior distribution to be chosen. However, this choice is not always obvious. Two different practitioners may sometimes disagree and there may be unforeseen consequences in a prior choice (see the 8-schools example in Gelman et al. [1995] or section 3 of Gelman [1996]). However, a completely subjective prior specification is challenging, and in practice, priors are often chosen to balance computational considerations with prior elicitation.

2.2 Nonparametric modelling

A statistical model consists of a set of probability measures on the sample space. In a parametric setting, the statistical model is assumed to be indexed or *parametrised*, by some finite set of parameters. However, in this setting we must ascertain whether the data generating distribution belongs to a parametric model. Often, we do not have such knowledge. Model selection approaches compare various parametric models through a trade-off between model complexity and goodness of fit.

Nonparametric models provide an alternative approach by removing the finite dimensional assumption of parametric models. Specifically, the number of parameters (model complexity) does not need to be specified *a priori* and is allowed to grow with the sample size. This is automatically inferred from a finite data set, with an additional benefit that although one may *a priori* believe that a population requires infinite parameters, a finite subset may only require a finite number.

A BNP model is a Bayesian model with an infinite dimensional parameter space. This allows for more flexible modelling, and consequently more reasonable inferences. Moreover, the prior acts as a penalty term to avoid over-fitting. However, constructing a prior distribution on an infinite dimensional parameter space is challenging and, in general, the basic criteria required are 1) large support 2) interpretable and easy to elicit hyperparameters and 3) tractable posterior inference.

2.3 Gaussian process

Definition 1. *A Gaussian process is a collection of random variables, such that any finite subset has a multivariate normal distribution with consistent parameters.*

Gaussian processes (GPs) are stochastic processes used for inferring non-linear and latent functions. They are defined as a family of normally-distributed random variables, indexed in this case by the input variable(s). In Bayesian inference, GPs are functionals, used as a prior probability distribution over function space, defined fully by a process mean and a symmetric positive definite covariance function. The latter of which is defined by a kernel function, which produces a Gram matrix when evaluated at the observed inputs. Kernel methods such as these are well-established tools for analysing the relationships between input data and corresponding outputs of complex functions. Kernels encapsulate the properties of a function in a computationally efficient manner. Additionally, they provide flexibility in terms of model complexity (the functions used to approximate the target function) through variation of the functional form and parameters of the kernel. An introduction of GPs is given in Rasmussen [2004].

These priors over function space excel when data is scarce or corrupted since they make strong *a priori* assumptions with regards to the relationship between datum and on the functions they learn. In making these assumptions, inference or optimisation can be performed over a reduced model space. This is in keeping with the 'no free lunch' theorem of computational complexity and optimisation, which states that the cost of inference/optimisation is the same for any method when averaged over all problems in that class. However, by using informed prior knowledge, one can choose a model which better matches the problem, (Wolpert and Macready [1997]).

Gaussian processes have found uses in numerous machine learning tasks, including supervised learning (where the objective is to learn relationships between inputs and outputs, e.g. regression and classification), unsupervised learning (where the objective is to learn the structure of a data set, e.g. manifold learning and dimensionality reduction), and reinforcement learning (where a goal is achieved by associating a positive action on an agent with a reward).

Gaussian processes can be explained from a number of perspectives, each of which are equivalent and reach the same result. Firstly, they can be introduced as an extension of a finite dimensional multivariate normal distribution to infinite dimensions by defining consistent finite dimensional Gaussian marginals, obtaining a stochastic process over a continuous and infinite indexing set. This is known

as the function space perspective as the distribution is specified directly on the unknown function. Alternatively, they can be introduced using weight spaces as a Bayesian generalisation to ridge regression and then extending further by projecting inputs into a higher dimensional space using a set of basis functions where linear relations can then be found. In this setting, the kernel is expressed as the inner product between the basis functions (known as the *kernel trick*), which is the basis of kernel methods. The GP is then obtained after marginalising over the weights. This is known as the weight space perspective, as the distribution is specified on the weights in the basis function expansion, which marginally leads to GP distribution on the function. For mathematical details of this perspective the reader is referred to Rasmussen [2004].

Prior over function space

In this section the function space perspective is presented, where a prior is specified directly on the unknown function. Consider a set of N observed covariates $\mathbf{X} \in \mathbb{R}^{N\times D}$ lying in a D-dimensional vector space $\mathcal{X}$, with corresponding vector $\mathbf{f} \in \mathbb{R}^N$of scalar function values. A mean function[1] is denoted $\mu : \mathbb{R}^D \to \mathbb{R}$ and symmetric positive definite covariance (kernel) function is denoted $K(\cdot,\cdot) : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$, with the consequent Gram matrix $K(\mathbf{X},\mathbf{X}) \in \mathbb{R}^{N\times N}$, of a real process $f(\mathbf{X})$ as:

$$\begin{aligned} \mu(\mathbf{X}) &= \mathbb{E}[f(\mathbf{X})], \\ K(\mathbf{X},\mathbf{X}') &= \mathbb{E}\left[(f(\mathbf{X}) - \mu(\mathbf{X}))(f(\mathbf{X}) - \mu(\mathbf{X}))^T\right], \end{aligned} \tag{2.1}$$

and write the GP prior over function space as:

$$\begin{aligned} \mathbf{f}|\mathbf{X} &\sim \mathcal{N}(\mu(\mathbf{X}), K(\mathbf{X},\mathbf{X})), \\ f(\cdot) &\sim \mathcal{GP}(\mu(\cdot), K(\cdot,\cdot)). \end{aligned} \tag{2.2}$$

where $\mathcal{X}$ is the indexing set of possible inputs to the Gaussian process. Here and throughout, $\mathrm{N}(\cdot,\cdot)$ denotes a normal distribution, in which the first argument is the mean vector and the second is the covariance matrix. Additionally $\mathrm{GP}(\cdot,\cdot)$ denotes a GP, in which the first argument is the mean function and the second is the covariance (kernel) function. A random Gaussian vector of function values can be generated by sampling the multivariate Gaussian distribution at a finite number of points in $\mathcal{X}$.

Consequently, two function values $f(\mathbf{X}_i)$ and $f(\mathbf{X}_j)$ evaluated at points in

[1] In many applications the process mean is zero and outputs are centred for simplicity.

the indexing set are jointly Gaussian with mean $\boldsymbol{\mu} = [\mu(\mathbf{X}_i), \mu(\mathbf{X}_j)]$ and covariance $[K_{ii}, K_{ij}; K_{ji}, K_{jj}]$, where the shorthand notation $K(\mathbf{X}_i, \mathbf{X}_j) = K_{ij}$ is used. The following two properties then apply. Given a random vector $\mathbf{A} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ in n-dimensional space then:

Property 1. *The marginal distribution is Gaussian. If we decompose* $\mathbf{A}$ *by splitting the finite indexing set into two disjoint subsets* $\{\{i\}, \{j\}\}$*, such that* $\mathbf{A} = [\mathbf{a}_i, \mathbf{a}_j]$*,* $\boldsymbol{\mu} = [\boldsymbol{\mu}_i, \boldsymbol{\mu}_j]$ *and* $\boldsymbol{\Sigma} = [\boldsymbol{\Sigma}_{ii}, \boldsymbol{\Sigma}_{ij}; \boldsymbol{\Sigma}_{ji}, \boldsymbol{\Sigma}_{jj}]$*, then* $\mathbf{a}_i \sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_{ii})$*, and* $\mathbf{a}_j \sim \mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_{jj})$*.*

Property 2. *The conditional distribution is Gaussian. Given the decomposition above,* $\mathbf{a}_i | \mathbf{a}_j \sim \mathcal{N}\left(\boldsymbol{\mu}_i + \boldsymbol{\Sigma}_{ij}\boldsymbol{\Sigma}_{jj}^{-1}(\mathbf{a}_j - \boldsymbol{\mu}_j), \boldsymbol{\Sigma}_{ii} - \boldsymbol{\Sigma}_{ij}^T\boldsymbol{\Sigma}_{jj}^{-1}\boldsymbol{\Sigma}_{ij}\right)$*.*

The existence of the GP is obtained from the Kolmogorov Extension Theorem (as an extension from the a consistent collection of finite dimensional distributions to a stochastic process) and the marginalisation property of the Gaussian distribution. Having defined this Gaussian process prior over function space, it is then possible to make inferences on the distribution over functions conditioned on a training set. The GP implies a joint Gaussian prior distribution over the function between training points and an unseen point:

$$\begin{bmatrix} \mathbf{f} \\ f(\mathbf{x}) \end{bmatrix} \sim \mathcal{N}\left(\left[\mu(\mathbf{X})^T, \mu(\mathbf{x})\right]^T, \begin{bmatrix} K(\mathbf{X}, \mathbf{X}), & K(\mathbf{X}, \mathbf{x})^T \\ K(\mathbf{X}, \mathbf{x}), & K(\mathbf{x}, \mathbf{x}) \end{bmatrix}\right) \tag{2.3}$$

Following from the conditional property 2, we can easily obtain the analytic conditional distribution of the test function value f, at $\mathbf{x}$ given observations $\mathbf{f}$:

$$\begin{aligned} f(\mathbf{x})|\mathbf{f} &\sim \mathcal{N}(m_1(\mathbf{x}), c_1(\mathbf{x})) \\ m_1(\mathbf{x}) &= \mu(\mathbf{x}) + K(\mathbf{x}, \mathbf{X})K(\mathbf{X}, \mathbf{X})^{-1}(\mathbf{f} - \mu(\mathbf{X})) \\ c_1(\mathbf{x}) &= K(\mathbf{x}, \mathbf{x}) - K(\mathbf{x}, \mathbf{X})K(\mathbf{X}, \mathbf{X})^{-1}K(\mathbf{x}, \mathbf{X})^T \end{aligned} \tag{2.4}$$

with Gaussian process posterior over function values:

$$f(\cdot)|\mathbf{f} \sim \mathcal{GP}(m_1(\cdot), c_1(\cdot)), \tag{2.5}$$

Function values $f(\mathbf{x})$ can then be sampled from the conditional distribution. Without loss of generality we can also follow this approach for multi dimensional outputs, leading to the *joint conditional distribution.*

The likelihood

In most modelling scenarios the true underlying function values are not known, but instead their noise corrupted values $\mathbf{y} = f(\mathbf{X}) + \boldsymbol{\eta}$. In this case the standard approach is to assume that the noise is additive, independent and identically Gaussian distributed, $\boldsymbol{\eta} \sim \mathcal{N}\left(\mathbf{0}, \beta^{-1}\mathbf{I}_N\right)$, leading to a factorised Gaussian likelihood:

$$p(\mathbf{y}|\mathbf{f}) = \mathcal{N}\left(\mathbf{y}|\mathbf{f}, \beta^{-1}\mathbf{I}_N\right) = \prod_{n=1}^{N} \mathcal{N}\left(y_n|f_n, \beta^{-1}\right), \tag{2.6}$$

Gaussian marginal likelihood, and Gaussian process marginal model:

$$\begin{aligned} p(\mathbf{y}|\mathbf{X}) &= \int p(\mathbf{y}|\mathbf{f})\, p(\mathbf{f}|\mathbf{X})\, d\mathbf{f} \\ y(\cdot) &\sim \mathcal{GP}\left(\mu(\cdot), K(\cdot,\cdot) + \beta^{-1}\delta_{(\cdot,\cdot)}\right), \end{aligned} \tag{2.7}$$

where $p(\mathbf{y}|\mathbf{X})$ is the marginal likelihood, $p(\mathbf{y}|\mathbf{f})$ is the likelihood, $p(\mathbf{f}|\mathbf{X})$ is the prior and $\beta^{-1}\delta_{(\cdot,\cdot)}$ is as a white noise kernel (also known as a *nugget* in this context). This kernel is a Kronecker-delta function scaled by a positive constant. An element of this kernel is then equal to the constant if both arguments are equal and zero otherwise. With this Gaussian likelihood, an analytic posterior GP over functions (and subsequent predictive distribution) is obtained:

$$\begin{aligned} f(\cdot)|\mathbf{y} &\sim \mathcal{GP}(m_2(\cdot), c_2(\cdot)) \\ f(\mathbf{x})|\mathbf{y} &\sim \mathcal{N}(m_2(\mathbf{x}), c_2(\mathbf{x})) \\ m_2(\cdot) &= \mu(\cdot) + K(\cdot,\mathbf{X})\left[K(\mathbf{X},\mathbf{X})^{-1} + \beta^{-1}I_N\right](\mathbf{y} - \mu(\mathbf{X})) \\ c_2(\cdot) &= K(\cdot,\cdot) - K(\cdot,\mathbf{X})\left[K(\mathbf{X},\mathbf{X})^{-1} + \beta^{-1}I_N\right]K(\cdot,\mathbf{X})^T \end{aligned} \tag{2.8}$$

and predictive posterior process over outputs, given new $\mathbf{x}$:

$$\begin{aligned} y(\cdot)|\mathbf{y} &\sim \mathcal{GP}\left(m_2(\cdot), c_2(\cdot) + \beta^{-1}\right) \\ y(\mathbf{x})|\mathbf{y} &\sim \mathcal{N}\left(m_2(\mathbf{x}), c_2(\mathbf{x}) + \beta^{-1}\right) \end{aligned} \tag{2.9}$$

The Gaussian process leads to tractable posterior inference under the assumption of a Gaussian likelihood of the outputs; while the latter is not always true, due to the interpretability and natural occurrence of the Gaussian distribution through the Central Limit Theorem, it is often justified.

Alternative likelihoods may also be used. For example in binary $(0, 1)$ classification, where the objective is to predict the probability that $y = 1$, one could transform the latent function through sigmoid (logistic), cumulative normal (probit) or (robust) threshold likelihood. Another example is count data for non-negative and discrete values, where latent functions are transformed to ensure positive support and then used as a rate parameter in a Poisson distribution.

When a GP prior is coupled with a non-Gaussian likelihood, an analytically tractable marginal likelihood or posterior process over outputs is no longer available. Stochastic approximations (such as Markov chain Monte Carlo (MCMC)), or deterministic approximations of integrals (such as Expectation Propagation, Laplace approximation or Variational approximation) must then be used to obtain an approximation to the non-Gaussian joint posterior process of latent functions.

The Laplace approximation is fast, but gives a poor approximation if the mode does not well describe the posterior (for example when using the Bernoulli probit likelihood). Expectation Propagation works very well under certain likelihoods and allows for sparse approximations but is otherwise slow, requires the ability to match moments and there can be convergence problems for some likelihoods. Variational methods can also give sparse approximations and give a principled approximation by optimizing a measure of divergence between the approximation and true distribution. However, these approximations often require factorisation assumptions to avoid a high dimensional integral. A comparison between different deterministic approximations and MCMC methods for GP classification can be found in Nickisch and Rasmussen [2008]. From this point only GPs with Gaussian likelihoods are discussed, where posterior inference is analytically tractable.

2.3.1 Kernels

The mean and covariance (kernel) of the stochastic process encapsulate prior assumptions on the latent function form. The kernel is a function that maps pairs of inputs to the positive real line, usually based on distance and conditional on a set of *hyperparameters* $\boldsymbol{\theta}$.

Kernels measure correlations between the unknown function at any pair of inputs (in the indexing set) in a higher dimensional, possibly infinite, *implicit* feature space, allowing for arbitrarily complicated functions. Conveniently, methods that use kernels do not need to calculate (potentially infinite) co-ordinates of data points in the implicit feature space. Instead they must only compute the kernel as the inner product of feature maps (the images of pairs of datum in the intrinsic feature

space):

$$K\left(\mathbf{x}, \mathbf{x}'\right) = \langle \phi\left(\mathbf{x}\right), \phi\left(\mathbf{x}'\right) \rangle. \tag{2.10}$$

Consequently the feature map must exist for a kernel to be valid. Mercer's theorem gives a principled approach to ensure the existence of these feature maps for a kernel - satisfying Mercer's condition ensures a kernel's Gram matrix is positive-semi definite, which is enough to ensure existence (Cortes and Vapnik [1995]). This process of replacing inner products in the input space with a kernel representing the inner product in the feature space (the *kernel trick*) allows for flexible and computationally efficient measures of correlation.

The covariance between evaluations of the unknown function at any pair of inputs in a GP is measured using kernels. Valid kernels lead to symmetric, semi-positive definite covariances. The kernel choice (including hyperparameters) constrain the family of functions which can be modelled. Consequently the kernel encodes our prior belief of the function to be modelled. When using a GP, the choice of kernel has a significant impact. For example, the squared exponential kernel (otherwise known as the *exponential quadratic* or *radial basis function*) makes strong assumptions of the function's smoothness. Using this kernel, the covariance between the ith and jth sample is given by:

$$K\left(\mathbf{x}_i, \mathbf{x}_j | \sigma, l\right) = \sigma^2 \exp\left\{ -\frac{1}{2l} \sum_{d=1}^{D} \left(x_{i,d} - x_{j,d}\right)^2 \right\}, \tag{2.11}$$

where σ denotes the signal variance controlling the average distance of the function away from its mean, and l denotes the lengthscale controlling how fast the function changes with respect to changes in the input[2]. This kernel is infinitely differentiable, and consequently only suitable for learning very smooth functions. A number of alternative kernels and a detailed discussion of their construction and application is given in Duvenaud [2014]. The periodic kernel is useful when the function repeats itself, but may not result in a flexible model, and a Matérn kernel is useful when we expect less smooth functions given it is $\lfloor \nu \rfloor$-times differentiable, where ν is a smoothness parameter.

We can increase the flexibility of these kernels by allowing each input dimension its own lengthscale. In doing this an *automatic relevance determination* (ARD) effect occurs (MacKay [1994], Rasmussen [2004]), where less relevant dimensions have larger lengthscales, and consequently the covariance depends more on the distance between more significant covariates. Further flexibility can be obtained by

[2]In Martingale theory, the lengthscale is directly related to the number of upcrossings.

using a combination of kernels. Common operators on kernels include:

1. Addition. This can be seen as an *OR* operator, where two input vectors are correlated if they are similar by either kernel. [3]

2. Multiplication. This can be seen as a *AND* operator, where two vectors are correlated if they are similar by both kernels.

Similarity is quantified by a measure of distance between covariates. In both cases, each kernel may be a function of all or a subset of covariates, where each kernel has a different form. Commonly a white noise kernel is added in this way. This models the assumption that data is corrupted by random fluctuations (such as measurement error), and inclusion of this noise term prevents over-fitting and ensures our covariances are positive definite (invertible). However, it is often equivalent to include a *nugget*[4], which is a scaled identity matrix, to the kernel through a likelihood.

2.4 Dirichlet process

Dirichlet processes (DPs) are a family of stochastic processes whose realisations are discrete probability measures. The parameters of the DP consist of a base distribution H (the expected value of the process) and a concentration parameter α (otherwise known as the scaling or mass parameter). Whilst the base distribution may be continuous, distributions drawn from the DP are almost surely (with probability one) discrete and the concentration parameter determines the strength of belief in the base distribution, with the process degenerating to the base measure as the concentration parameter approaches infinity. The formal definition of the DP is:

Definition 2. *Given a measurable space S, a base probability distribution H_0 on S and a positive real number α, the Dirichlet process $DP(\alpha, H_0)$ is a stochastic process whose realization (i.e. a sample drawn from the process) is a probability distribution over S, and such that for any measurable finite partition of S, denoted $\{B_i\}_{i=1}^n$, if $H \sim DP(\alpha, H_0)$, then $(H(B_1), \ldots, H(B_n)) \sim Dir(\alpha H_0(B_1), \ldots, \alpha H_0(B_n))$.*

Dirichlet processes were introduced by Ferguson [1973a] who used the Kolmogorov Consistency Theorem to show their existence and described the Dirichlet

[3] For a GP this decomposition is maintained through Bayes rule, obtaining a posterior which can also be expressed as a sum of kernels.

[4] This naming convention has roots in geo-statistics where Gaussian process regression (then referred to as Kriging) was first applied.

process posterior using Bayes rules and the conjugacy between the Dirichlet and multinomial distributions. Specifically, assuming

$$H \sim \mathrm{DP}\left(\alpha, H_0\right) \text{ and } \theta_i | H \overset{iid}{\sim} H \text{ for } i = 1, \ldots, n, \tag{2.12}$$

conditioned the n observations this posterior is:

$$H|\theta_{1:n} \sim \mathrm{DP}\left(\alpha + n, \frac{\alpha H_0 + \sum_{i=1}^{n} \delta_{\theta_i}}{\alpha + n}\right) \tag{2.13}$$

For explanation of this posterior process the reader is referred to Ferguson [1973a]. However, as realisations of the DP are discrete almost surely, it is common to convolute a known or specified density (that is absolutely continuous with respect to some measure Λ) with the DP to produce realisations that are absolutely continuous (with respect to Λ) almost surely Lo [1984]. This induces an infinite mixture model for flexible density estimation by making use of the DP as a prior for the unknown mixing measure. This approach is used in chapter 5. In this setting, the posterior distribution is no longer analytically tractable, however the hierarchical structure of the model can be used for stochastic or deterministic approximations.

Other mathematically equivalent representations of the DP have emerged which vary by their perspectives. Blackwell and MacQueen [1973] used de Finetti's theorem to prove existence and introduced the Blackwell-MacQueen Pólya urn scheme which characterizes the marginal law of the exchangeable sequence $(\theta_1, \theta_2, \ldots)$, see also Pitman [1996]. In Sethuraman [1994b], a way of constructing a DP was introduced using a stick-breaking construction and finally Aldous [1985] introduced the Chinese restaurant process construction. Each of these are briefly outlined in the following sections.

2.4.1 Blackwell-MacQueen

This representation draws motivation from the Pólya urn model, in which we have an urn containing balls of various colours. The model then proceeds by randomly selecting a ball, replacing it and adding an additional ball of the same colour. This sampling, in a 'rich get richer' fashion, underpins the Blackwell-MacQueen construction.

Assuming the existence of the Dirichlet process, the Blackwell-MacQueen scheme can now be presented, which describes the sequence of predictive distributions of $(\theta_1, \theta_2, \ldots)$ obtained from (2.12) after marginalising over the random probability measure H. To begin, consider an empty urn and a base distribution

H_0 representing the prior belief of the distribution over colours. The first ball of colour θ_1 is sampled from H_0 and added to the urn, and the sampling process for θ_n then proceeds as:

1. With probability proportional to α, draw $\theta_n \sim H_0$ and add a ball of this colour in the urn.

2. With probability proportional to $n-1$, draw a random ball i for $i = 1, \ldots, n-1$ from the urn, observe its colour θ_i, and place it back to the urn and add an additional ball of the same colour $\theta_n = \theta_i$ in the urn.

This produces a sequence of samples $\theta_1, \theta_2, \ldots$ with predictive distributions:

$$\theta_n | \theta_{1:n-1} \sim \frac{\alpha H_0 + \sum_{i=1}^{n-1} \delta_{\theta_i}}{\alpha + n - 1}. \tag{2.14}$$

Conversely, Blackwell and MacQueen [1973] start with the sequence of predictive distribution in (2.14) and show that the sequence $(\theta_1, \theta_2, \ldots)$ is exchangeable, i.e for any $n \in \mathbb{N}$, the joint distribution of the finite sequence is invariant to any finite permutation of the indices. Consequently by using the de Finettis theorem there must exist a distribution over a random probability measure such that, conditioned on this random probability measure, the sequence is independent and identically distributed, and this distribution is the Dirichlet process. As a result, it is proven that the Blackwell-MacQueen urn scheme is a representation of the DP and a way to construct it is obtained.

2.4.2 Chinese restaurant process

The Chinese restaurant process can be directly linked to the Blackwell-MacQueen urn scheme. We assume that there is a Chinese restaurant with an infinite number of tables. As the customers enter the restaurant they sit randomly to any of the occupied tables or they choose to sit at an empty table.

The process defines a distribution on the space of partitions of the positive integers. We start by drawing $\theta_1, \ldots, \theta_n$ from the Blackwell-MacQueen urn scheme, where θ_i may duplicate resulting in a clustering. These define a partition of the set $\{1, 2, \ldots, n\}$ in k clusters. Consequently drawing from the Blackwell-MacQueen urn scheme induces a random partition of the set. The Chinese restaurant process is this induced distribution over partitions. Starting with one customer on the first table:

1. With probability $\frac{\alpha}{\alpha+n}$ the $n+1$ customer sits at an unoccupied table.

2. With probability $\frac{c_j}{\alpha+n}$ the $n+1$ customer sits on the jth occupied table, where c_j is the number of people sitting on that table and $\sum_{j=1}^{k} c_j = n$.

We use this construction throughout.

2.4.3 Stick-breaking

Draws from a DP are composed of a weighted sum of (countably infinite) point masses. Here a construction for sampling the distribution H in this way is presented. The weights are obtained via *stick-breaking*; given a stick of length 1, break off a proportion β_1 and assign π_1 equal to broken piece's length. Repeat the same process on the remaining length of stick to obtain $\pi_2, \pi_3, \ldots$; due to the way that this scheme is defined the process can be repeated infinitely many times.

Based on the above the π_i can be modelled as $\pi_i = \beta_i \prod_{j=1}^{i-1} (1 - \beta_j)$, where the $\beta_i \sim \text{Beta}(1, \alpha)$, while the atoms of the point masses are sampled directly from the base distribution $\theta_i^* \sim H_0$. Consequently H can be written as a sum of delta functions weighted with π_i probabilities which is equal to:

$$H \sim \sum_{i=1}^{\infty} \pi_i \delta_{\theta_i^*} \tag{2.15}$$

Thus the stick-breaking construction gives a simple and intuitive way to construct a Dirichlet process.

2.5 Bayesian inference

So far we have introduced prior processes over both function spaces (in the form of the GP prior), and over distributions (in the form of the DP prior). Using Bayes rule to update the beliefs in these process models led to a posterior distribution for the processes, which were analytically tractable in conjugate settings.

However, in many practical settings, one must move beyond the conjugate setting, for example, to non-normal likelihoods in GP-based models or mixture models in DP-based models. Additionally, hyper-priors may be placed on the hyper-parameters in our models to express their uncertainty. This leads to difficulties when calculating the normalization factor. When this happens an alternative is a stochastic approximation (such as MCMC), which allows posterior inference through sampling that is exact up to Monte Carlo error, providing convergence guarantees. The literature on MCMC methods is well developed, and an extensive introduction

is given in Brooks et al. [2011], and an introduction to its application in Machine Learning is given in Andrieu et al. [2003].

Alternatives to the stochastic approximations given by MCMC are provided by deterministic approximations such as the Laplace approximation, variational approximation, and expectation propagation, which are often much faster but lack the convergence guarantees. In this thesis, we focus on MCMC algorithms and variational approximation. The following subsections give a brief discussion of each MCMC method used within this thesis and on variational inference.

2.5.1 Metropolis-Hastings algorithm

The Metropolis-Hastings (MH) algorithm was first conceptualised in Metropolis et al. [1953] for the numerical calculation of the equation of state for a system of rigid spheres. Instead of choosing system configurations randomly and then weighting by their Boltzmann factor, configurations were chosen with probability equal to the Boltzmann factor and weighted evenly. The sampling scheme which followed is the MH algorithm. Whilst the method originated for specific problems in numerical simulations of physical systems, the scope of applications now covers the entire of computational science (Hastings [1970]).

The Metropolis-Hastings algorithm is particularly useful in the Bayesian analysis setting, where the posterior distribution is known up to a factor of proportionality (the marginal normalization factor). Despite missing this, the MH algorithm can draw samples without the need to calculate the factor, which is often extremely difficult in practice. An introduction to MH in the Bayesian setting can be found in Chib and Greenberg [1995] and Robert and Casella [1999]. Transitions between states of the Markov Chain are governed by a *proposal distribution* $q\left(\tilde{x}|x\right)$ (also known as *conditional density* or *candidate kernel*) and the unnormalised (in this case posterior) *target distribution* to be sampled $\pi\left(\cdot\right)$. The procedure is outlined in Algorithm 1. The tilde circumflex denotes a proposed parameter, and the superscript gives the Markov Chain state.

Whilst, here, the proposal distribution's parametrisation is chosen *a priori*, an alternative is to implement an adaptive Metropolis-Hastings algorithm in which the proposal distribution for state i may depend on the previous sampled states, $\theta^{(1)}, \ldots, \theta^{(i-1)}$. An example is provided in Haario et al. [2001], where the proposal covariance matrix is tuned at each step to target a scaled version of the posterior covariance matrix, based on the sample covariance matrix of the sampled states $\theta^{(1)}, \ldots, \theta^{(i-1)}$.

Algorithm 1 Metropolis-Hastings algorithm with symmetric proposal distribution

Initialise $\theta^{(0)}$.
for $i = 1, 2, \ldots$ **do**
  Propose: $\tilde{\theta} \sim q\left(\theta^{(i)}|\theta^{(i-1)}\right)$
  Calculate acceptance probability:

$$\alpha\left(\tilde{\theta}|\theta^{(i-1)}\right) = \min\left\{1, \frac{q\left(\theta^{(i-1)}|\tilde{\theta}\right)}{q\left(\tilde{\theta}|\theta^{(i-1)}\right)} \frac{\pi\left(\tilde{\theta}\right)}{\pi\left(\theta^{(i-1)}\right)}\right\}$$

  Sample $u \sim \text{Uniform}\,(0, 1)$
  if $u < \alpha\left(\tilde{\theta}|\theta^{(i-1)}\right)$ **then**
    Accept proposal: $\theta^{(i)} \leftarrow \tilde{\theta}$
  else
    Reject proposal: $\theta^{(i)} \leftarrow \theta^{(i-1)}$.
  end if
end for

2.5.2 Hamiltonian Monte Carlo

Another Metropolis algorithm is Hamiltonian Monte Carlo, otherwise known as Hybrid Monte Carlo (HMC), (Duane et al. [1987], Neal et al. [2011], Betancourt [2017] and Betancourt et al. [2017]). Unlike the Metropolis-Hastings algorithm, HMC is an auxiliary variable sampler that uses the gradient based Hamiltonian evolution to reduce the correlation between successive sampled states. Consequently, HMC targets states with a higher acceptance criteria. This causes it to converge to the target probability distribution quicker and with less random walk behaviour.

The Hamiltonian is defined as an energy function in terms of a position vector $\mathbf{q}(t)$ and a momentum vector $\mathbf{p}(t)$ at time t: $H\left(\mathbf{q}(t), \mathbf{p}(t)\right) = E_U(\mathbf{q}(t)) + E_K(\mathbf{p}(t))$, where $E_U(\mathbf{q})$ is the potential energy and $E_K(\mathbf{p})$ is the kinetic energy, the sum of which is constant. The evolution of this system is then defined by the partial derivatives of the Hamiltonian:

$$\frac{d\mathbf{p}}{dt} = -\frac{\partial H}{\partial \mathbf{q}}, \quad \frac{d\mathbf{q}}{dt} = +\frac{\partial H}{\partial \mathbf{p}}. \tag{2.16}$$

The potential energy is defined as the negative log probability density of the target distribution, with an additive constant chosen for convenience. In the case of posterior inference this leads to:

$$E_U(\mathbf{q}(t)) = -\log\left(\text{likelihood}\left(\mathbf{q}(t)\right)\right) - \log\left(\text{prior}\left(\mathbf{q}\left(t\right)\right)\right).$$

Furthermore, it is convention to define the kinetic energy as:

$$E_K(\mathbf{p}(t)) = \frac{1}{2}\mathbf{p}(t)\mathbf{M}_K^{-1}\mathbf{p}(t), \tag{2.17}$$

where $\mathbf{M}_K$ is a symmetric, positive definite mass matrix, chosen to be a scalar multiple of the identity matrix. Hamiltonian dynamics describe an object's motion in continuous time, but to simulate the dynamics numerically the Hamiltonian equations must be approximated by discretising time. This is achieved by splitting the interval on which the dynamics are simulated into smaller intervals of fixed length δ, and using an iterative solver such as Euler's method or the leap frog method. The procedure using the leap frog method is given in Algorithm 2.

Algorithm 2 Hamiltonian Monte Carlo algorithm (leap frog method)

Initialise $\theta^{(0)} = q\,(t_0)$.
for $i = 1, 2, \ldots$ **do**
Take a half step $\delta/2$ to update the momentum variable:

$$\mathbf{p}\,(t_{i-1} + \delta/2) = \mathbf{p}\,(t_{i-1}) - (\delta/2)\,\frac{\partial E_U}{\partial \mathbf{q}\,(t_{i-1})}$$

Take a full step δ to update the position variable:

$$\theta^{(i)} = \mathbf{q}\,(t_{i-1} + \delta) = \mathbf{q}\,(t_{i-1}) + \delta\frac{\partial E_K}{\partial \mathbf{p}\,(t_{i-1} + \delta/2)}$$

Take another half step δ to update the momentum variable:

$$\mathbf{p}\,(t_{i-1} + \delta) = \mathbf{p}\,(t_{i-1} + \delta/2) - (\delta/2)\,\frac{\partial E_U}{\partial \mathbf{q}\,(t_{i-1} + \delta)}$$

Update discretised time step $t_i = t_{i-1} + \delta$
end for

2.5.3 Elliptical slice sampling

Slice sampling is another auxiliary variable sampler which samples from a (univariate) density by introducing additional slice variables. Conditioned on the slice, the method requires sampling uniformly from intervals with density above the slice. In practice, this is done adaptively by making proposals inside a bracket which shrinks automatically until the point lies within the slice.

Elliptical slice sampling is another Metropolis auxiliary variable sampler for performing inference in models with multivariate Gaussian priors (Murray et al. [2010]). These priors often occur in many probabilistic models and are usually

associated with strong dependencies between parameters and/or latent variables of the model.

Inference in these models can rarely be performed in closed form and, consequently, a deterministic or stochastic approximation of the posterior must often be applied. Elliptical slice sampling is a generalisation of the Metropolis-Hasting algorithm with a Gaussian proposal distribution and fixed step size, chosen *a priori*. Elliptical slice sampling generalises this by allowing the step size to vary, defining a locus of proposals which intersect the current state θ. Moreover, it combines this proposal with slice sampling to produce a rejection-free sampler. An equivalent definition of this proposal is:

$$\tilde{\theta} = \nu \sin(\alpha) + \theta \cos(\alpha), \quad \nu \sim \mathcal{N}(0, \Sigma), \tag{2.18}$$

where Σ is the covariance of a zero-mean Gaussian prior on θ, and in which adjustments of α are synonymous with adjusting step size. Elliptical slice sampling is a simple and generic algorithm which applies to many models, working well for a variety of GP based models. It benefits from requiring no tuning parameters and is rejection-free. These properties make the method ideal for use while model building, removing the need to spend time deriving and tuning updates for more complex algorithms. The procedure is outlined in Algorithm 3.

2.5.4 Gibbs sampling

Another method, sometimes used in conjunction with the previous methods, is Gibbs sampling. This was first conceptualised in Geman and Geman [1984], and gains its namesake from Josiah Gibbs, following an analogy to statistical mechanics. The algorithm is useful when a joint distribution cannot be sampled directly, but sampling the conditional distributions of each variable, or sets of variables, is possible. Gibbs sampling can be interpreted as a Metropolis method with a sequence of proposals in the form of conditional distributions, which are always accepted. The procedure is outlined in Algorithm 4, where the subscript gives the hyperparameter dimension, and D is the number of hyperparameters.

Algorithm 3 Elliptical slice sampling algorithm

Initialise $\theta^{(0)}$.
for $i = 1, 2, \ldots$ **do**
  Choose an ellipse: $\nu \sim \mathcal{N}(0, \Sigma)$
  Obtain a log-likelihood threshold:

$$u \sim \text{Uniform}(0, 1)$$
$$\log y \leftarrow \log\left(\text{Likelihood}\left(\theta^{(i-1)}\right)\right) + \log(u)$$

  Draw step size and define bracket:

$$\alpha \sim \text{Uniform}(0, 1)$$
$$[\alpha_{\min}, \alpha_{\max}] \leftarrow [\alpha - 2\pi, \alpha]$$

  while $\theta^{(i)}$ not set **do**
    Propose new state: $\tilde{\theta} \leftarrow \theta^{(i-1)} \cos\alpha + \nu \sin\alpha$
    if $\log\left(\text{Likelihood}\left(\tilde{\theta}\right)\right) > \log y$ **then** $\theta^{(i)} \leftarrow \tilde{\theta}$, break.
    else shrink bracket and draw step size:
      if $\alpha < 0$ **then** $\alpha_{\min} \leftarrow \alpha$ **else** $\alpha_{\max} \leftarrow \alpha$ **end if**
      $\alpha \sim \text{Uniform}(\alpha_{\min}, \alpha_{\max})$
    end if
  end while
end for

Algorithm 4 Gibbs sampling algorithm

Initialise $\boldsymbol{\theta}^{(0)} = \{\theta_1^{(0)}, \ldots, \theta_D^{(0)}\} \in \mathbb{R}^D$.
for $i = 1, 2, \ldots$ **do**
  for $d = 1, \ldots, D$ **do**
    Sample $\theta_d^{(i)} \sim p\left(\theta_d | \theta_1^{(i)}, \ldots, \theta_{d-1}^{(i)}, \theta_{d+1}^{(i-1)}, \ldots, \theta_D^{(i-1)}\right)$
  end for
end for

Gibbs sampling is known to perform poorly in models with strong dependencies between variables (Titsias et al. [2009]), due to high autocorrelation in the chain and slow mixing.

2.5.5 Pseudo-marginal Monte Carlo

When a marginal likelihood cannot be evaluated pointwise, Bayesian inference becomes even more challenging. The results of Andrieu and Roberts [2009] and Beaumont [2003] reveal that it is possible to use an unbiased estimate of the marginal

likelihood to sample from the correct posterior target distribution. The result has been applied to many types of Metropolis algorithms, including: pseudo-marginal MH (Andrieu and Roberts [2009]); pseudo-marginal HMC (Lindsten and Doucet [2016]); and pseudo-marginal slice sampling (Murray and Graham [2016]).

Suppose the unnormalised target distribution of section 2.5.1 is a marginal distribution, defined through the integral:

$$\pi(\theta) = \int \pi'(\theta, z)\, dz,$$

where z is a latent random variable. In the case of posterior inference:

$$\begin{aligned} \pi(\theta) &= p(\theta|Y) \propto p(Y|\theta)\, p(\theta) \\ \pi'(\theta, z) &= p(\theta, z|Y) \propto p(Y|\theta, z)\, p(\theta, z) \end{aligned} \tag{2.19}$$

When this marginalisation is intractable, an approach is to use MCMC methods, typically a Gibbs sampler in combination with those listed above to sample from the joint posterior, and subsequently use the samples of θ to study the target $\pi(\theta)$. However, there exists a multitude of scenarios where this is neither feasible nor practical. For example, it may not be possible to simulate the latent variables (they may be infinite dimensional objects). Alternatively the latent variables may be high dimensional, or the correlations between variables may be large, resulting in poor mixing.

The pseudo-marginal Monte Carlo (PMMC) approach instead approximates the marginal density $\pi(\theta)$, required for calculation of the acceptance probability for a Metropolis transition operator, with an estimator $\hat{f}(\theta)$ which can be evaluated pointwise, is non-negative everywhere, and is unbiased:

$$\mathbb{E}\left[\hat{f}(\theta)\right] = \pi(\theta). \tag{2.20}$$

A common approach to obtain this estimator is importance sampling. This requires an importance density $q_\theta(z)$ (otherwise known as biased, proposal, or sample distribution[5]) that can be sampled to obtain:

$$\hat{f}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \frac{\pi'(\theta, z_i)}{q_\theta(z_i)}, \qquad z_i \overset{iid}{\sim} q_\theta(\cdot) \tag{2.21}$$

Given this estimator, the PMMC procedure follows Algorithm 5

[5]It is also required that $\{z : q_\theta(z) > 0\} \supset \{z : \pi'(\theta, z) > 0\}$.

Algorithm 5 Pseudo-marginal Metropolis Hastings algorithm with symmetric proposal distribution

Initialise $\theta^{(0)}$.
for $i = 1, 2, \ldots$ **do**
  Propose: $\tilde{\theta} \sim q\left(\theta^{(i)}|\theta^{(i-1)}\right)$
  Compute pseudo-marginal $\hat{f}\left(\tilde{\theta}\right)$.
  Calculate acceptance probability:

$$\alpha\left(\tilde{\theta}|\theta^{(i-1)}\right) = \min\left\{1, \frac{q\left(\theta^{(i-1)}|\tilde{\theta}\right)}{q\left(\tilde{\theta}|\theta^{(i-1)}\right)} \frac{\hat{f}\left(\tilde{\theta}\right)}{\hat{f}\left(\theta^{(i-1)}\right)}\right\}$$

  Sample $u \sim \text{Uniform}(0, 1)$
  if $u < \alpha\left(\tilde{\theta}|\theta^{(i-1)}\right)$ **then**
    Accept proposal: $\theta^{(i)} \leftarrow \tilde{\theta}$
  else
    Reject proposal: $\theta^{(i)} \leftarrow \theta^{(i-1)}$.
  end if
end for

2.5.6 Variational inference

Rather than sample the distributions arising from intractable integrals using stochastic MCMC sampling, deterministic approximations of the distribution can be obtained at the expense of asymptotic convergence guarantees. This is usually achieved using optimisation and consequently these methods are more readily able to use distributed optimisation (Welling and Teh [2011] andAhmed et al. [2012]), or stochastic optimisation (Robbins and Monro [1951] and Kushner and Yin [1997]), which results in faster inference and makes them more suitable for larger data sets.

One such approach to deterministic approximation is variational inference (VI), a method from machine learning which uses optimisation to approximate probability densities Blei et al. [2017]. Variational inference posits a family of densities and then finds the member of that family which is closest to the target distribution according to the Kullback-Leibler (KL) divergence measure[6], otherwise known as relative entropy. Formally, the reverse KL divergence between two distributions q and p is defined as:

$$D_{\text{KL}}(q||p) = \int_{-\infty}^{\infty} q(x) \log \frac{q(x)}{p(x)} dx \tag{2.22}$$

[6]This is an asymmetric information-theoretic distance measure between two densities.

In a Bayesian setting there are two popular variational strategies: variational Bayes, which assumes a factorised variational posterior and variational expectation maximisation (VEM). This section focusses on the latter, which is utilised in chapter 3. In this setting, the posterior distribution of the latent variables z conditioned on hyperparameters θ, denoted $\pi(z|\theta)$ is approximated by a variational distribution $q(z|\theta)$ belonging to a family of densities $\mathcal{Q}$ which is chosen *a priori*. The objective is to find:

$$q^*(z|\theta) = \underset{q(z|\theta)\in\mathcal{Q}}{\arg\min} D_{\mathrm{KL}}(q(z|\theta)\,||\pi(z|\theta)), \tag{2.23}$$

while simultaneously optimising over θ. However, due to the marginal density of the observations (otherwise called *evidence*) $\log p(Y|\theta)$, this is typically not analytically tractable. Consequently, this may be reformulated into a problem of maximising the evidence lower bound (ELBO) which is equivalent to minimising the KL divergence up to the evidence term which is constant with respect $q(z|\theta)$ for fixed θ:

$$\begin{aligned}
\mathrm{ELBO}_\theta(q) &= \int q(z|\theta)\log p(z,Y|\theta)\,dz - \int q(z|\theta)\log q(z|\theta)\,dz \\
&= \int q(z|\theta)\left[\log p(z|\theta) + \log p(Y|z,\theta) - \log q(z|\theta)\right]dz \\
&= \int q(z|\theta)\log p(Y|z,\theta) - D_{\mathrm{KL}}(q(z|\theta)\,||p(z|\theta)).
\end{aligned} \tag{2.24}$$

Optimizing the ELBO with respect to q (the E-step) and θ (the M-step) is conceptually similar to the expectation maximisation algorithm. Here the first term is the expected log likelihood, and maximising this term encourages variational densities that better explain the observed data. The second term encourages densities close to the prior. Consequently it can be seen that this variational objective function mirrors the balance between likelihood and prior of Bayesian approaches.

Using the definitions of ELBO, the KL divergence, and that $D_{\mathrm{KL}}(\cdot,\cdot) \geq 0$, a lower-bound for the log evidence is observed:

$$\log p(Y|\theta) = D_{\mathrm{KL}}(q(z|\theta)\,||\pi(z|\theta)) + \mathrm{ELBO}_\theta(q) \tag{2.25}$$

$$\geq \mathrm{ELBO}_\theta(q). \tag{2.26}$$

This lower bound can also be found using Jensen's inequality for concave functions (such as the log function) Jordan et al. [1999]. It must be noted that optimal values

of θ maximise a lower bound to the marginal likelihood, not necessarily the marginal likelihood and thus are only approximate maximum (marginal) likelihood values.

Chapter 3

Bayesian inference for the Gaussian process latent variable model

The GPLVM introduced by Lawrence [2004] is a hierarchical model originally used for unsupervised learning tasks for non-linear dimension reduction where inputs are not directly observed. The model treats these inputs as unobserved latent variables and places independent GP priors over the mapping from latent to output space. In Lawrence [2005] a Gaussian prior is placed on the latent variables, which are optimised to the *maximum a posteriori* (MAP) solution (equivalent to maximum likelihood (ML) with L2 regularisation). To capture uncertainty in the latent variables Titsias and Lawrence [2010] developed a variational method for GPLVMs. The GPLVM may be extended to the supervised learning case (sGPLVM) where latent points indexed by known and observable inputs are obtained from GP mapping from the observed input space to the latent space.

A novel framework for fully Bayesian inference of the supervised GPLVM is introduced in this chapter. This is motivated by the need to quantify hyper-parameter uncertainty, and more accurately quantify latent uncertainty. Dependent on choice of divergence, variational methods necessarily under or over-estimate the variance. When correlations between approximated variables (within variational factorisations) are large, and when these factors are over a non-trivial number of dimensions this approximation becomes increasingly poor. This is demonstrated on a simple example based on simulated data, while also demonstrating the benefits of fully Bayesian inference on predictive performance when compared with VEM using the projected process approximation. Moreover, it sheds a light on situations

when the approximate maximum marginal likelihood (MML) estimates, obtained from optimising a lower bound to the marginal likelihood, are poor.

This chapter begins with an introduction to the sparse Gaussian processes from existing literature, which builds the foundations to introduce the sGPLVM model. Following this, a novel framework which overcomes high correlations between latent variables and hyperparameters is presented. This is achieved by using an unbiased pseudo-estimate for the marginal likelihood that approximately integrates over the latent variables. This is used to construct a Markov Chain to explore the hyperparameters posterior. The Gibbs sampler can then be uncollapsed, sampling latent variables using elliptical slice sampling. This approach obtains Markov chains of posterior samples that are guaranteed to converge asymptotically to the true target distribution.

3.1 Review

In this section sparse methods for GPs and the GPLVM are reviewed.

3.1.1 Variational sparse Gaussian process

Whilst the GP formulation outlined in section 2.3 leads to nonparametric and data driven models of significant flexibility, it also results in a model with memory and computational limitations from the need to store and invert the kernel, which comes with a computational complexity of $\mathcal{O}\left(N^3\right)$. Consequently, without modification, working with larger data sets can be infeasible. For this reason a number of methods for scaling up GPs have been developed, and are often referred to as sparse Gaussian processes. This nomenclature stems from the first developments, which focussed on using *sparse* subsets of the data set to approximate the kernel. In contrast, later methods focus on model or posterior approximations.

In Snelson and Ghahramani [2006a], the set of selected *sparse* points were generalised so they did not need to be subsets of the data. This expands the probability space by introducing m pairs of *pseudo* (auxiliary) inputs and outputs, which reduce the computational complexity to $\mathcal{O}\left(N \times M^2\right)$, where $M \ll N$ and N is the number of samples. The objective is to then find a good low-rank approximation of the kernel and the optimal choice of inducing point locations using gradient based optimisation of the marginal likelihood. This method was given the name Fully Independent Training Conditional in Quiñonero-Candela and Rasmussen [2005], where this and other related approximate methods of the time are reviewed. In order to compare the methods, the authors showed that these mod-

els can be seen as a modifications of the GP prior over functions. This leads to significant increase in flexibility, where *pseudo-points* can be considered as model hyperparameters. However, optimising over them can lead to over-fitting (see Bauer et al. [2016] and de Garis Matthews [2016]). Additionally, there is no measure of distance between the exact (full) model and the modified (sparse) model. This can lead to numerous difficulties with model validation (see Naish-Guzman and Holden [2008]).

Variational inference is another method used to scale up GPs through approximate inference. With recent developments, variational methods have become an extremely powerful tools for Bayesian inference. A summary of recent developments is given in Zhang et al. [2017]. This is used in Csató and Opper [2002] and Seeger et al. [2003] where *pseudo-points* are treated as model parameters. Instead Titsias [2009] presented a *variational sparse Gaussian process*, based on the variational approximation to the posterior process in which *pseudo-points* are treated as variational parameters and learnt alongside model hyperparameters by maximising the evidence lower bound (ELBO) of the log-marginal likelihood. This new outlook comes with a number of benefits: the approximation is nonparametric, therefore predictions away from the data take the same form as true posterior; the approximation monotonically improves as the number of *pseudo-points* increases; optimisation of *pseudo-points* comes down to maximising the ELBO and regularisation in the objective function naturally avoids over-fitting *variational* parameters; only the approximate posterior GP must be evaluated to make predictions which does not require additional steps or further approximation[1].

3.1.2 Gaussian process latent variable models

The GPLVM extends the application of GPs to an unsupervised learning task where the objective is to learn the underlying structure of the data by learning a non-linear manifold (Lawrence [2004] and Lawrence [2005]). This model can be considered a non-linear generalisation to the *dual* of probabilistic principal component analysis (PPCA)[2], where dual refers to optimisation being performed over unobserved latent variables $\mathbf{z}$ and *linear* transformations $\mathbf{W}$ being marginalised, instead of the vice versa in PPCA. The linear relationship between latent and observed variables, with

[1] This prediction is equivalent to that of the Projected Process approximation (otherwise known as Deterministic Training Conditional), another method which can be seen as a modification of the Gaussian process prior over functions (Quiñonero-Candela and Rasmussen [2005]) but has a tendency to over-fit.

[2] Similar to probabilistic principal component analysis, principal component analysis is obtained as a limiting case when noise converges to 0, for a particular linear kernel. The generalisation to non-linear manifold learning is achieved by the choice of the kernel.

additive noise is then given by:

$$\mathbf{y}_n = \mathbf{W}\mathbf{z}_n + \boldsymbol{\eta}_n, \qquad \text{for } n = 1, \ldots, N, \tag{3.1}$$

where k_y is the dimension of observed variable $\mathbf{y}_n$, k_z is the dimension of the latent variable $\mathbf{z}_n$, with $k_z \ll k_y$, N is the sample size and $\mathbf{W} \in \mathbb{R}^{k_y \times k_z}$. Specifying the priors over transformations and noise gives the probability model:

$$\begin{aligned} p(\boldsymbol{\eta}_n) &= \mathcal{N}\left(\mathbf{0}, \beta^{-1}\mathbf{I}_{k_y}\right), \quad p(\mathbf{W}) = \prod_{d=1}^{k_y} \mathcal{N}\left(\mathbf{w}_d | \mathbf{0}, \mathbf{I}_{k_z}\right) \\ p(\mathbf{y}_n | \mathbf{W}, \mathbf{z}_n, \beta) &= \mathcal{N}\left(\mathbf{y}_n | \mathbf{W}\mathbf{z}_n, \beta^{-1}\mathbf{I}_{k_y}\right) \end{aligned} \tag{3.2}$$

where $\mathbf{w}_d$ is the dth row of matrix $\mathbf{W}$. Marginalising over the linear projection matrix gives the marginal likelihood:

$$\begin{aligned} p(\mathbf{Y}|\mathbf{Z}, \beta) &= \int \prod_{d=1}^{k_y} p(\mathbf{y}_{:,d}|\mathbf{Z}, \mathbf{W}, \beta)\, p(\mathbf{W})\, d\mathbf{W} \\ &= \prod_{d=1}^{k_y} \mathcal{N}\left(\mathbf{y}_{:,d} | \mathbf{0}, \mathbf{Z}\mathbf{Z}^T + \beta^{-1}\mathbf{I}_N\right) \end{aligned} \tag{3.3}$$

This conjugate prior leads to a product of Gaussian distributions with a linear covariance kernel. This *dual* probabilistic principal component model can then be generalised for non-linear manifold learning tasks by replacing the linear kernel for one which measures non-linear correlations, and consequently models non-linear projection functions. Point estimates of the latent variables are then obtained by maximising this marginal likelihood with L2 regularisation using gradients, which are analytically available for many kernel choices. This model is now presented in terms of the GP distributed mapping from latent to output space.

Using the notation $f_{n,d} = f_d(\mathbf{z}_n)$, the latent function values are defined by a matrix $\mathbf{F} \in \mathbb{R}^{N \times k_y}$. Additionally independent GP prior distributions are defined across features:

$$p(\mathbf{F}|\mathbf{Z}, \boldsymbol{\theta}) = \prod_{d=1}^{k_y} p(\mathbf{f}_{:,d}|\mathbf{Z}, \boldsymbol{\theta}), \tag{3.4}$$

with:

$$p(\mathbf{f}_{:,d}|\mathbf{Z}, \boldsymbol{\theta}) = \mathcal{N}\left(\mathbf{f}_{:,d} | \mathbf{0}, \mathbf{K}_\mathbf{f}\right), \tag{3.5}$$

in which $\mathbf{K}_\mathbf{f} \in \mathbf{R}^{N \times N}$ is a kernel (covariance) matrix, the n, n'-th entry of which is

$K_f(\mathbf{z}_n, \mathbf{z}_{n'}; \boldsymbol{\theta})$ where the set of hyperparameters in the kernel is denoted $\boldsymbol{\theta}$. Thus, coupled with a Gaussian likelihood, the marginal likelihood is given by:

$$\begin{aligned} p(\mathbf{Y}|\mathbf{Z}, \boldsymbol{\theta}, \beta) &= \int \prod_{d=1}^{k_y} \prod_{n=1}^{N} p(y_{n,d}|f_{n,d}, \beta)\, p(\mathbf{f}_{:,d}|\mathbf{Z}, \boldsymbol{\theta})\, d\mathbf{F} \\ &= \prod_{d=1}^{k_y} \mathcal{N}\left(\mathbf{y}_{:,d}|\mathbf{0}, \mathbf{K_f} + \beta^{-1}\mathbf{I}_N\right), \end{aligned} \tag{3.6}$$

3.2 Supervised Gaussian process latent variable model

To extend the GPLVM to the supervised case a prior is placed on the latent points, again in the form of independent GP priors $z_d(\mathbf{x}) \sim \mathrm{GP}(0, K_z(\mathbf{x}, \mathbf{x}'; \boldsymbol{\sigma}))$, $d = 1, \ldots, k_z$, denoting the set of kernel hyperparameters $\boldsymbol{\sigma}$. Thus:

$$p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\sigma}) = \prod_{d=1}^{k_z} p(\mathbf{z}_{:,d}|\mathbf{X}, \boldsymbol{\sigma}) = \prod_{d=1}^{k_z} \mathcal{N}(\mathbf{z}_{:,d}|\mathbf{0}, \mathbf{K_z}), \tag{3.7}$$

where $\mathbf{K_z} \in \mathbb{R}^{N \times N}$ is the kernel matrix, with n, n'-th entry equal to $K_z(\mathbf{x}_n, \mathbf{x}_{n'}; \boldsymbol{\sigma})$. The joint probability density over the observed data and latent variables is:

$$p(\mathbf{Y}, \mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\sigma}, \beta) = p(\mathbf{Y}|\mathbf{Z}, \boldsymbol{\theta}, \beta)\, p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\sigma}) \tag{3.8}$$

The latent function values $\mathbf{F}$ of the GPs have already been marginalised, (3.6).

3.2.1 Variational marginalisation of latent variables.

This model was studied in a dynamic setting by Damianou et al. [2011]. It can further be viewed as a deep GP model (Damianou and Lawrence [2013]) with a single hidden layer.

In many Bayesian models, posterior inference may be sensitive to the choice of hyperparameters; this includes GP-based models, such as the GPLVM, where the choice of kernel hyperparameters can have a large impact on inferences. There are generally two approaches to overcome this (in the absence of strong prior knowledge): hierarchical Bayes, with a hyper-prior assigned to account for uncertainty in the hyperparameters, and empirical Bayes, which plug-in estimates of the hyperparameters. Typically in empirical Bayes these estimates are taken to maximise the marginal likelihood. However, for the GPLVM, computing the marginal likelihood requires integration with respect to the latent variables, which is analytically intractable as they appear non-linearly in the inverse kernel matrix. A major ad-

vancement was provided in the work of Titsias and Lawrence [2010], who developed a VEM approach, assuming a Gaussian variational posterior and utilising sparse GPs to obtain a closed form lower bound to the marginal likelihood. In an expectation maximisation fashion, this lower bound can then be optimised with respect to the hyperparameters to obtain approximate type II maximum marginal likelihood estimates. This can be generalised to the supervised case, studied in this thesis, and described fully below.

Considering the E-step of the VEM algorithm in isolation, the basic idea consists of using a proxy variational distribution (including variational hyper-parameters) over the latent variables in order to approximate the posterior distribution. The variational parameters of this distribution are chosen to minimise the KL divergence between the proxy distribution and the posterior. It is well known that this choice of divergence often tends to underestimate the variance; this is particularly true when the posterior is highly correlated but the proxy distribution has a factorised form, or when the posterior is a mixture of Gaussians with well separated modes and the proxy is a single Gaussian. However, the reverse may also be true. For example, when approximating a mixture of Gaussians with poorly separated modes with a single Gaussian (see Turner and Sahani [2011] for more details). In the case of the GPLVM, the posterior (conditioned on the hyperparameters) can be sampled exactly with ESS, and these samples can be used to understand the quality of a variational posterior; this is discussed further in section 3.3.

It is first noted that standard mean field variational methodologies (as previously used in PPCA and Factor Analysis models (Bishop [1999] and Jordan et al. [1999]) do not lead to an analytically tractable algorithm. Instead, the variational distribution is restricted to lie within a class. Specifically, consider a variational distribution $q(\mathbf{Z})$, which is taken to have the following factorised Gaussian form:

$$q(\mathbf{Z}) = \prod_{d=1}^{k_z} \mathcal{N}(\mathbf{z}_{:,d} | \boldsymbol{\mu}_d, \mathbf{S}_d), \tag{3.9}$$

where $\mathbf{S}_d$ is a diagonal $N \times N$ covariance matrix and conditional dependence on $\mathbf{X}$ and hyperparameters $(\boldsymbol{\sigma}, \boldsymbol{\theta}, \beta)$ enters through optimization of the variational parameters $\boldsymbol{\mu}_d \in \mathbb{R}^N$ and $\mathbf{S}_d \in \mathbb{R}^{N \times N}$. Using Jensen's inequality the ELBO can be

derived as:

$$\begin{aligned}\log p\left(\mathbf{Y}|\mathbf{X},\boldsymbol{\theta},\boldsymbol{\sigma},\beta\right) &= \log\left[\int p\left(\mathbf{Y}|\mathbf{Z},\boldsymbol{\theta},\beta\right)p\left(\mathbf{Z}|\mathbf{X},\boldsymbol{\sigma}\right)d\mathbf{Z}\right] \\ &= \log\left[\int p\left(\mathbf{Y}|\mathbf{Z},\boldsymbol{\theta},\beta\right)\frac{p\left(\mathbf{Z}|\mathbf{X},\boldsymbol{\sigma}\right)}{q\left(\mathbf{Z}\right)}q\left(\mathbf{Z}\right)\right]d\mathbf{Z} \\ &\geq \int q\left(\mathbf{Z}\right)\log p\left(\mathbf{Y}|\mathbf{Z},\boldsymbol{\theta},\beta\right)d\mathbf{Z} - \int q\left(\mathbf{Z}\right)\log\frac{q\left(\mathbf{Z}\right)}{p\left(\mathbf{Z}|\mathbf{X},\boldsymbol{\sigma}\right)}d\mathbf{Z} \\ &:= \tilde{\mathcal{F}}\left(q\left(\mathbf{Z}\right),\boldsymbol{\theta},\beta\right) - \mathrm{KL}\left(q\left(\mathbf{Z}\right)||p\left(\mathbf{Z}|\mathbf{X},\boldsymbol{\sigma}\right)\right).\end{aligned} \tag{3.10}$$

Given that the data $\{\mathbf{y}_i\}_{i=1}^N$ is independent across features the first term can be expanded as follows:

$$\tilde{\mathcal{F}}\left(q\left(\mathbf{Z}\right),\boldsymbol{\theta},\beta\right) = \sum_{d=1}^{k_y}\int q\left(\mathbf{Z}\right)\log p\left(\mathbf{y}_{:,d}|\mathbf{Z},\boldsymbol{\theta},\beta\right)d\mathbf{Z} := \sum_{d=1}^{k_y}\tilde{\mathcal{F}}_d\left(q\left(\mathbf{Z}\right),\boldsymbol{\theta},\beta\right). \tag{3.11}$$

The second term is the negative KL divergence between two Gaussian distributions and can, therefore, be evaluated with ease. The term $\tilde{F}_d\left(q\left(\mathbf{Z}\right),\boldsymbol{\theta},\beta\right)$, however, is clearly still analytically intractable as $\mathbf{Z}$ remains inside the inverse of the kernel. In order to formulate a tractable problem, the variational sparse GP approach of Titsias [2009] is applied, which augments the probability model with inducing variables. For each vector of latent function values $\mathbf{f}_{:,d}$, $d = 1,\ldots,k_z$, a separate set of M auxiliary *inducing variables* $\mathbf{u}_{:,d} \in \mathbb{R}^M$ is introduced, and are evaluated at a set of M inducing points given by the matrix $\mathbf{Z_u} \in \mathbb{R}^{M\times k_z}$. The inducing points are independent of the training points. The inducing variables are simply function values drawn from the GP prior on $\mathbf{f}_{:,d}$ (common across d). For simplicity, all of the $\mathbf{u}_{:,d}$ are evaluated at the same inducing locations. With these inducing variables the augmented probability model is as follows:

$$p\left(\mathbf{y}_{:,d},\mathbf{f}_{:,d},\mathbf{u}_{:,d}|\mathbf{Z},\boldsymbol{\theta},\beta,\mathbf{Z_u}\right) = p\left(\mathbf{y}_{:,d}|\mathbf{f}_{:,d},\beta\right)p\left(\mathbf{f}_{:,i}|\mathbf{Z},\boldsymbol{\theta},\mathbf{u}_{:,d},\mathbf{Z_u}\right)p\left(\mathbf{u}_{:,d}|\mathbf{Z_u},\boldsymbol{\theta}\right),$$

since the joint GP prior over $\mathbf{f}_{:,d}$ and $\mathbf{u}_{:,d}$ evaluated at $\mathbf{Z},\boldsymbol{\theta}$ and $\mathbf{Z_u}$ factorizes, with conditional Gaussian prior:

$$p\left(\mathbf{f}_{:,d}|\mathbf{Z},\boldsymbol{\theta},\mathbf{u}_{:,d},\mathbf{Z_u}\right) = \mathcal{N}\left(\mathbf{f}_{:,d}|\boldsymbol{\alpha}_d,\mathbf{K_f} - \mathbf{K_{fu}}\mathbf{K_u}^{-1}\mathbf{K_{uf}}\right), \tag{3.12}$$

in which $\mathbf{K_u}$ is the covariance matrix corresponding to the inducing points, $\mathbf{K_{fu}} = \mathbf{K_{uf}}^T$ is the cross-covariance between the inducing and the latent points and $\boldsymbol{\alpha}_d =$

$\mathbf{K_{fu}}\mathbf{K_u}^{-1}\mathbf{u}_{:,d}$. The Gaussian prior over the inducing variables is $p\left(\mathbf{u}_{:,d}|\mathbf{Z_u},\boldsymbol{\theta}\right) = \mathcal{N}\left(\mathbf{u}_{:,d}|\mathbf{0},\mathbf{K_u}\right)$. By marginalising out $(\mathbf{f}_{:,d},\mathbf{u}_{:,d})$ the likelihood $p\left(\mathbf{y}_{:,d}|\mathbf{Z},\boldsymbol{\theta},\beta\right)$ can then be found. This is true for any set of inducing points $\mathbf{Z_u}$ and, consequently, they can be considered variational parameters, rather than random variables or hyperparameters. This is an important distinction as optimisation over variational parameters is less prone to over-fitting. However, this does not make the model immune to this, as optimisation must still be performed over hyperparameters.

From this point onwards notation is simplified by dropping the dependence on $\mathbf{Z_u}$ in expressions. Variational inference must now be applied a second time to approximate the true posterior $p\left(\mathbf{f}_{:,d}|\mathbf{u}_{:,d},\mathbf{y}_{:,d},\mathbf{Z},\boldsymbol{\theta},\beta\right)p\left(\mathbf{u}_{:,d}|\mathbf{y}_{:,d},\mathbf{Z},\boldsymbol{\theta},\beta\right)$, using the sparse variational distribution:

$$q\left(\mathbf{f}_{:,d},\mathbf{u}_{:,d}\right) = p\left(\mathbf{f}_{:,d}|\mathbf{u}_{:,d},\mathbf{Z},\boldsymbol{\theta}\right)\phi\left(\mathbf{u}_{:,d}\right),$$

where $p\left(\mathbf{f}_{:,d}|\mathbf{u}_{:,d},\mathbf{Z},\boldsymbol{\theta}\right)$ is the conditional GP prior given in (3.12) and $\phi\left(\mathbf{u}_{:,d}\right)$ is the variational distribution over inducing variables. The lower bound of the log likelihood term in the integrand of $\tilde{\mathcal{F}}_d$ in (3.11) are given by:

$$\begin{aligned}\log p\left(\mathbf{y}_{:,d}|\mathbf{Z},\boldsymbol{\theta},\beta\right) \geq & \int \phi\left(\mathbf{u}_{:,d}|\boldsymbol{\theta}\right)\log\frac{p\left(\mathbf{u}_{:,d}\right)\mathcal{N}\left(\mathbf{y}_{:,d}|\boldsymbol{\alpha}_i,\beta^{-1}I_N\right)}{\phi\left(\mathbf{u}_{:,d}\right)}d\mathbf{u}_{:,d} \\ & -\frac{\beta}{2}\operatorname{Tr}\left(\mathbf{K_f}-\mathbf{K_{fu}}\mathbf{K_u}^{-1}\mathbf{K_{uf}}\right).\end{aligned} \tag{3.13}$$

In contrast to Titsias [2009], it is necessary to force independence of the distribution $\phi\left(\mathbf{u}_{:,d}\right)$ from $\mathbf{Z}$. Combining the lower bounds above with (3.11) gives:

$$\begin{aligned}\tilde{\mathcal{F}}_d\left(q\left(\mathbf{Z}\right),\boldsymbol{\theta},\beta\right) \geq & \int q\left(\mathbf{Z}\right)\Big[\int\phi\left(\mathbf{u}_{:,d}\right)\log\frac{p\left(\mathbf{u}_{:,d}|\boldsymbol{\theta}\right)\mathcal{N}\left(\mathbf{y}_{:,d}|\boldsymbol{\alpha}_d,\beta^{-1}I_N\right)}{\phi\left(\mathbf{u}_{:,d}\right)}d\mathbf{u}_{:,d} \\ & -\frac{\beta}{2}\operatorname{Tr}\left(\mathbf{K_f}\right)+\frac{\beta}{2}\operatorname{Tr}\left(\mathbf{K_u}^{-1}\mathbf{K_{uf}}\mathbf{K_{fu}}\right)\Big]d\mathbf{Z},\end{aligned}$$

using the standard properties of the trace of a matrix. Under the factorisation assumption $\phi\left(\mathbf{u}_{:,d}\right)$ does not depend on $\mathbf{Z}$ and so the integrations can be interchanged:

$$\begin{aligned}\tilde{\mathcal{F}}_d\left(q\left(\mathbf{Z}\right),\boldsymbol{\theta},\beta\right) \geq & \int\phi\left(\mathbf{u}_{:,d}\right)\left[\left\langle\log\mathcal{N}\left(\mathbf{y}_{:,d}|\boldsymbol{\alpha}_d,\beta^{-1}I_N\right)\right\rangle_{q(\mathbf{Z})}+\log\frac{p\left(\mathbf{u}_{:,d}|\boldsymbol{\theta}\right)}{\phi\left(\mathbf{u}_{:,d}\right)}\right]d\mathbf{u}_{:,d} \\ & -\frac{\beta}{2}\operatorname{Tr}\left(\left\langle\mathbf{K_f}\right\rangle_{q(\mathbf{Z})}\right)+\frac{\beta}{2}\operatorname{Tr}\left(\mathbf{K_u}^{-1}\left\langle\mathbf{K_{uf}}\mathbf{K_{fu}}\right\rangle_{q(\mathbf{Z})}\right),\end{aligned}$$

where $\left\langle f(z)\right\rangle_{q(z)}$ denotes the expectation of $f\left(z\right)$ under $q\left(z\right)$. Now the lower bound under the distribution $\phi\left(\mathbf{u}_{:,d}\right)$ can be maximised analytically. The optimal setting

of this distribution is:

$$\phi\left(\mathbf{u}_{:,d}\right) \propto e^{\left\langle \log \mathrm{N}\left(\mathbf{y}_{:,d}|\boldsymbol{\alpha}_d, \beta^{-1} I_N\right)\right\rangle_{q(\mathbf{Z})}} p\left(\mathbf{u}_{:,d}|\boldsymbol{\theta}\right)$$

and the lower bound that incorporates such an optimal setting is obtained by inserting $\phi\left(\mathbf{u}_{:,d}\right)$ into the lower bound expression:

$$\begin{aligned}\tilde{\mathcal{F}}_d\left(q\left(\mathbf{Z}\right), \boldsymbol{\theta}, \beta\right) \geq & \log\left(\int e^{\left\langle \log \mathrm{N}\left(\mathbf{y}_{:,d}|\boldsymbol{\alpha}_d, \beta^{-1} I_N\right)\right\rangle_{q(\mathbf{Z})}} p\left(\mathbf{u}_{:,d}|\boldsymbol{\theta}\right) d\mathbf{u}_{:,d}\right) \\ & - \frac{\beta}{2} \operatorname{Tr}\left(\left\langle \mathbf{K}_{\mathbf{f}} \right\rangle_{q(\mathbf{Z})}\right) + \frac{\beta}{2} \operatorname{Tr}\left(\mathbf{K}_{\mathbf{u}}^{-1} \left\langle \mathbf{K}_{\mathbf{uf}} \mathbf{K}_{\mathbf{fu}} \right\rangle_{q(\mathbf{Z})}\right).\end{aligned} \tag{3.14}$$

For a number of kernels this can now be computed in closed form. Optimisation may now be performed on the tractable variational lower bound according to (3.14), using Scaled Conjugate Gradients with respect to the variational parameters $\left(\{\boldsymbol{\mu}_d, \mathbf{S}_d\}_{d=1}^D, \mathbf{Z}_{\mathbf{u}}\right)$ and hyperparameters $(\boldsymbol{\theta}, \boldsymbol{\sigma}, \beta)$ to obtained approximate ML estimates.

Remark 1. *Following Damianou [2015], within this chapter* $\left(\{\boldsymbol{\mu}_d, \mathbf{S}_d\}_{d=1}^D, \mathbf{Z}_{\mathbf{u}}\right)$ *are treated as free parameters, and optimised directly with scaled conjugate gradients alongside the model hyperparameters, using a re-parametrisation. This approach mitigates against local optima, but does not aid against other problems associated with optimisation of hyperparameters. This is implemented using SheffieldML [2017].*

The analytic computations can be found in Titsias and Lawrence [2010] and the gradient derivations can be found in Damianou [2015], alongside derivations for the predictive density:

$$p\left(\mathbf{Y}_*|\mathbf{Y}\right) \approx \int p\left(\mathbf{Y}_*|\mathbf{F}_*\right) q\left(\mathbf{F}_*|\mathbf{Z}_*\right) q\left(\mathbf{Z}_*\right) d\mathbf{Z}_* d\mathbf{F}_*, \tag{3.15}$$

where $q\left(\mathbf{Z}_*\right)$ is obtained using standard GP regression, and $q\left(\mathbf{F}_*|\mathbf{Z}_*\right)$ is expressed as a product of terms with the same form as the projected process approximation. This integral is a non-Gaussian multivariate density that cannot be computed. Consequently, the variational scheme instead computes the first and second moments which are available in closed-form. However, in order to sample and evaluate the predictive distribution this is assumed to be a multi-variate Gaussian with corresponding first and second moments in this thesis.

3.3 Pseudo-marginal Monte Carlo for the GPLVM

This section introduces a novel framework for fully Bayesian inference of the supervised GPLVM. Model hyperparameters defining the covariance function have important implications for smoothness, complexity, and relevance of the inputs. It is common to optimise these parameters based on the approximate MML, known as approximate type II maximum likelihood, using gradient-based optimisation. However, the likelihood as a function of these parameters is non-convex, and consequently practitioners often find that the optimisation is highly dependent on initialisation, with no guarantee of a satisfactory local optimum (Bitzer and Williams [2010]). This is particularly profound when the data set is small or has a low signal-to-noise ratio, which is often the motivation for using Bayesian approaches.

Moreover, it must be emphasised that hyperparameter estimates in section 3.2.1 do not optimise the marginal likelihood, but a lower bound to the marginal likelihood. The consequences of this in simple examples was shown in Turner and Sahani [2011]. Specifically, they found that it is not important for the lower bound to be as tight as possible to the marginal likelihood, but that it is equally tight everywhere. If this is not the case, the effect is to push estimates away from peaks in the likelihood and towards regions where the bound is tighter. Another interesting conclusion is that biases in the hyperparameter estimates increase considerably as the number of hyperparameters increases.

Additionally, while variational methods can substantially reduce computational time, this comes at the cost of strong assumptions and considerable bias. For example, there are often assumptions of independence, on the forms of distributions and, dependent upon the choice of divergence, variational methods underestimate or overestimate the variance of these distributions (Blei et al. [2017]). In particular, the simulated examples of section 3.3.4 compare the variational posterior on the latent variables with the true posterior samples obtained from ESS (conditional on hyperparameter values). In this setting, a large underestimation of the variance is found in some cases.

This motivates a Bayesian framework for inference with the sGPLVM introduced in section 3.2. This Bayesian approach naturally regularises against overfitting by penalising unnecessary model complexity. Moreover, an understanding of uncertainty in the hyperparameters is gained, alongside sound uncertainty quantification in predictions by integrating over the hyperparameters. The framework overcomes the high correlations between latent variables and hyperparameters by using an unbiased pseudo estimate for the marginal likelihood that approximately integrates

over the latent variables in a collapsed Gibbs sampler. This is used to construct a Markov Chain to explore the posterior of the hyperparameters, and then these samples can be used alongside ESS to sample the latent variables. This overcomes issues with optimisation of the hyperparameters and avoids the distributional and independence assumptions of variational methods. This framework is referred to as 'pseudo-marginal' (PM) throughout this thesis. The procedure is demonstrated on simulated examples, showing the improved quantification of uncertainty and multimodality of the hyperparameters, and improved UQ in predictions when compared with those obtained using sGPLVM with the variational approach. Another important contribution is to shed light on situations when the variational scheme works well and when it is poor, by considering simulated scenarios that are increasingly miss-specified by the sGPLVM.

3.3.1 Collapsed pseudo-marginal Gibbs sampling

The natural choice to explore the posterior of the latent variables and hyperparameters is a Gibbs sampling algorithm, which alternates between sampling and fixing the latent variables and hyperparameters. In the GPLVM family of models the latent parameters and hyperparameters are strongly coupled, leading to sharp peaks in the posterior when latent variables are fixed. This results in poor MCMC mixing and slow convergence rates (Filippone and Girolami [2014]) and a method that can break these correlations is required.

Although analytical integration of the latent variables $\mathbf{Z}$ is intractable since they appear non-linearly in the inverse kernel matrix $\mathbf{K_f}$, the correlation between the latent variables and hyperparameters can be broken by approximately integrating over the latent variables through a PMMC scheme. The results of Andrieu and Roberts [2009] and Beaumont [2003] reveal that an unbiased estimate of the marginal likelihood can be used to sample from the correct hyperparameter posterior distribution.

Within this chapter importance sampling is used to obtain the unbiased approximation to the marginal likelihood based on the approximate distribution $q(\mathbf{Z}) \approx p(\mathbf{Z}|\mathbf{Y}, \mathbf{X}, \boldsymbol{\sigma}, \boldsymbol{\theta}, \beta)$, which is known as the *proposal, biased or sampling distribution*. Drawing Q importance samples, the unbiased estimate of the marginal is:

$$\tilde{p}(\mathbf{Y}|\mathbf{X}, \boldsymbol{\sigma}, \boldsymbol{\theta}, \beta) \simeq \frac{1}{Q} \sum_{q=1}^{Q} \frac{p(\mathbf{Y}|\mathbf{Z}^{(q)}, \boldsymbol{\theta}, \beta) p(\mathbf{Z}^{(q)} \mid \mathbf{X}, \boldsymbol{\sigma})}{q(\mathbf{Z}^{(q)})}, \tag{3.16}$$

where $\mathbf{Z}^{(q)} \overset{iid}{\sim} q(\mathbf{Z})$, and $p(\mathbf{Y}|\mathbf{Z}^{(q)}, \boldsymbol{\theta}, \beta)$ and $p(\mathbf{Z}^{(q)}|\mathbf{X}, \boldsymbol{\sigma})$ are the GP models given by (3.6) and (3.7) respectively. For the proposal distribution $q(\mathbf{Z})$ the approximate variational posterior of section 3.2 is utilised. In this setting the hyperparameters $(\boldsymbol{\sigma}, \boldsymbol{\theta}, \beta)$ are fixed at the required sample and only the E-step of the variational scheme is performed, to optimise the lower bound with respect to the variational parameters. Importantly this avoids constraints on the tightness of the lower bound required to obtain good hyperparameter estimates. This pseudo-marginal can now be used to sample from the posterior of the hyperparameters in a Metropolis algorithm.

To improve mixing, the set of hyperparameters $\boldsymbol{\xi} = (\boldsymbol{\sigma}, \boldsymbol{\theta}, \beta)$ are split into R adjoint subsets, $\boldsymbol{\xi}_r$, $r = 1, \ldots, R$. The full conditionals of each block can then be sampled in a Metropolis-Hastings within Gibbs algorithm. Each block $\boldsymbol{\xi}_r$ is updated with a random walk based on a transformation $\boldsymbol{\eta}_r = t_r(\boldsymbol{\xi}_r)$ to ensure full support on the real space of appropriate dimension, and a multivariate normal proposal distribution is used for the transformed parameter: $\pi(\boldsymbol{\eta}_r'|\boldsymbol{\eta}_r) \sim \mathrm{N}(\mathbf{0}, \boldsymbol{\Sigma}_r)$. This gives the proposal distribution $\pi(\boldsymbol{\xi}_r) = |\partial t_r/\partial \boldsymbol{\xi}_r|\pi(\boldsymbol{\eta}_r)$ in the original parameter space. The acceptance probability for a move from $\boldsymbol{\xi}_r$ to $\boldsymbol{\xi}_r'$ is therefore:

$$\tilde{\alpha}(\boldsymbol{\xi}_r, \boldsymbol{\xi}_r') = \min\left[1, \frac{\tilde{p}(\mathbf{Y}|\mathbf{X}, \boldsymbol{\xi}')p(\boldsymbol{\xi}_r')}{\tilde{p}(\mathbf{Y}|\mathbf{X}, \boldsymbol{\xi})p(\boldsymbol{\xi}_r)} \frac{|\partial t_r/\partial \boldsymbol{\xi}_r(\boldsymbol{\xi}_r)|}{|\partial t_r/\partial \boldsymbol{\xi}_r(\boldsymbol{\xi}_r')|}\right],$$

In addition a variant of the adaptive Metropolis-Hastings algorithm of Haario et al. [2001] is employed, which adapts the proposal covariance matrix $\boldsymbol{\Sigma}_r$ to approximate the target distribution's covariance matrix multiplied by a constant s_{d_r}. Following Haario et al. [2001], this constant is chosen to be $s_{d_r} = 2.38^2/d_r$ where d_r is the dimension of the block. The algorithm then begins with an initial proposal covariance matrix for each block, and after g_0 iterations this is updated using the sample covariance, with a small positive constant on the diagonal. The full procedure is outlined in Algorithm 6.

Algorithm 6 Pseudo-marginal adaptive MH in Gibbs.

for $g = 1, 2, \ldots$ **do**

Set $\boldsymbol{\xi}^{(g)} = \boldsymbol{\xi}^{(g-1)}$

for each $\boldsymbol{\xi}_r$, $r = 1, \ldots, R$ **do**

Sample $\boldsymbol{\eta}_r' = \boldsymbol{\eta}_r^{(g)} + \boldsymbol{\epsilon}_g$ where $\boldsymbol{\epsilon}_g \sim \mathrm{N}(\mathbf{0}, \boldsymbol{\Sigma}_r^{(g-1)})$.

Find the unbiased approximation $\tilde{p}(\mathbf{Y}|\mathbf{X}, \boldsymbol{\xi}')$ using importance sampling (3.16).

Set:

$$\boldsymbol{\xi}_r^{(g)} = \begin{cases} \boldsymbol{\xi}_r' & \text{with probability } \tilde{\alpha}(\boldsymbol{\xi}^{(g)}, \boldsymbol{\xi}') \\ \boldsymbol{\xi}_r^{(g-1)} & \text{with probability } 1 - \tilde{\alpha}(\boldsymbol{\xi}_r^{(g)}, \boldsymbol{\xi}') \end{cases}.$$

if $g > g_0$ **then**

$$\boldsymbol{\Sigma}_r^{(g)} = \frac{s_{d_r}}{g-1} \left[\sum_{m=1}^{g} \boldsymbol{\eta}_r^{(m)} \boldsymbol{\eta}_r^{(m)^T} - g \bar{\boldsymbol{\eta}}_r \bar{\boldsymbol{\eta}}_r^T \right] + s_{d_r} \epsilon \mathbf{I}.$$

end if

end for

return $\boldsymbol{\xi}^{(g)}$ for $g > n_0$.

end for

3.3.2 Uncollapsing with elliptical slice sampling

Samples of the latent variables can now be obtained using the ESS algorithm of Murray et al. [2010], given the hyperparameters sampled in the previous section. These samples will be used to compute predictions in section 3.3.3. The target distribution for the sampler is the full conditional of the latent variables:

$$\begin{aligned} p(\mathbf{Z}|\mathbf{Y}, \mathbf{X}, \boldsymbol{\xi}) &\propto p(\mathbf{Y}|\mathbf{Z}, \boldsymbol{\theta}, \beta)\, p(\mathbf{Z}|\boldsymbol{\sigma}, \mathbf{X}) \\ &\propto \prod_{d=1}^{k_y} \mathcal{N}\left(\mathbf{y}_{:,d}|\mathbf{0}, K_f(\mathbf{Z}, \mathbf{Z}; \boldsymbol{\theta}) + \beta^{-1}\mathbf{I}_N\right) \times \\ &\quad \prod_{d=1}^{k_z} \mathcal{N}\left(\mathbf{z}_{:,d}|0, K_z(\mathbf{X}, \mathbf{X}; \boldsymbol{\sigma})\right), \end{aligned}$$

and the proposal distribution is given by:

$$\mathbf{Z}' = \boldsymbol{\nu} \sin\alpha + \mathbf{Z} \cos\alpha, \qquad \boldsymbol{\nu}_{:,d} \overset{iid}{\sim} \mathcal{N}(0, \mathbf{K}_\mathbf{z}), \qquad d = 1, \ldots, k_z.$$

This defines a full ellipse passing through the previous state $\mathbf{Z}$ and a prior sample $\boldsymbol{\nu} \in \mathbb{R}^{N \times k_z}$ as α varies. This proposal depends on a tuning parameter α which would be chosen *a priori* under a normal Metropolis-Hastings scheme. The algorithm of Murray et al. [2010] adaptively chooses this tuning parameter using slice

sampling. The procedure for sampling $\mathbf{Z}$ using the elliptical slice sampler is given in Algorithm 7.

Algorithm 7 Elliptical slice sampler for the latent variables.

Require: current state $\mathbf{Z}$, and log-likelihood function.
Ensure: new state $\mathbf{Z}'$.

1: Sample: $\boldsymbol{\nu} \sim \prod_{d=1}^{k_z} \mathcal{N}(\boldsymbol{\nu}_{:,d}|0, \mathbf{K}_z)$, creating an ellipse at current state with $\alpha = 0$.
2: Log-likelihood threshold:

$$u \sim \text{Uniform}[0,1], \quad \log h \leftarrow \log p(\mathbf{Y}|\mathbf{Z}; \boldsymbol{\theta}, \beta) + \log u.$$

3: Draw an initial proposal, define bracket on the ellipse:

$$\alpha \sim \text{Uniform}[0, 2\pi], \quad [\alpha_{\min}, \alpha_{\max}] \leftarrow [\alpha - 2\pi, \alpha].$$

4: **while** not returned **do**
5: Propose new latent variables:

$$\mathbf{Z}' \leftarrow \boldsymbol{\nu} \sin\alpha + \mathbf{Z} \cos\alpha.$$

6: **if:** $\log p(\mathbf{Y}|\mathbf{Z}', \boldsymbol{\theta}, \beta) > \log h$ (proposal lies in slice) **then:**
7: Accept: **return** $\mathbf{Z}'$.
8: **else:**
9: Shrink bracket and re-sample step size:
10: **if** $\alpha < 0$ **then:** $\alpha_{\min} \leftarrow \alpha$ **else:** $\alpha_{\max} \leftarrow \alpha$
11: $\alpha \sim \text{Uniform}[\alpha_{\min}, \alpha_{\max}]$.
12: **end while**

3.3.3 Predictions using Markov Chain Monte Carlo

Predictions can now be made by marginalising over the posterior samples, without the need for distributional assumptions or point estimates. The marginalised predictive density for a test point $\mathbf{x}_*$ is:

$$p(\mathbf{y}_*|\mathbf{x}_*, \mathbf{Y}, \mathbf{X}) = \int p(\mathbf{y}_*|\mathbf{z}_*, \mathbf{Z}, \mathbf{Y}, \boldsymbol{\theta}, \beta) \times p(\mathbf{z}_*|\mathbf{x}_*, \mathbf{X}, \mathbf{Z}, \boldsymbol{\sigma})\, p(\mathbf{Z}, \boldsymbol{\xi}|\mathbf{Y}, \mathbf{X})\, d\mathbf{z}_* d\mathbf{Z} d\boldsymbol{\xi}. \tag{3.17}$$

The second term inside the integral of (3.17) is the predictive density of the latent variable $\mathbf{z}_*$ given the latent variables, hyperparameters and data, which is given by the GP predictive density:

$$p(\mathbf{z}_*|\mathbf{x}_*, \mathbf{X}, \mathbf{Z}, \boldsymbol{\sigma}) = \prod_{d=1}^{k_z} \mathcal{N}\left(z_{*d}|\mathbf{K}_{\mathbf{z}*}^T \mathbf{K}_{\mathbf{z}}^{-1} \mathbf{z}_{:,d}, s_*\right), \tag{3.18}$$

with $s_* = k_z(\mathbf{x}_*, \mathbf{x}_*; \boldsymbol{\sigma}) - \mathbf{K}_{\mathbf{z}*}^T \mathbf{K}_{\mathbf{z}}^{-1} \mathbf{K}_{\mathbf{z}*}$. Here $\mathbf{K}_{\mathbf{z}*}$ is the cross-covariance at the training inputs $\mathbf{X}$ and the test input $\mathbf{x}_*$. Similarly, the first term inside the integral of (3.17) is the predictive density of the test output $\mathbf{y}_*$ given $\mathbf{z}_*$, the latent variables, hyperparameters and data, which is given by the GP predictive density:

$$p\left(\mathbf{y}_* | \mathbf{z}_*, \mathbf{Z}, \mathbf{Y}, \boldsymbol{\theta}, \beta\right) = \prod_{d=1}^{k_y} \mathcal{N}\left(y_{*d} | \mathbf{A}\mathbf{y}_{:,d}, \mathbf{S} + \beta^{-1}\right),$$

where $\mathbf{A} = \mathbf{K}_{\mathbf{f}*}^T (\mathbf{K}_{\mathbf{f}} + \beta^{-1}\mathbf{I}_N)^{-1}$ and $\mathbf{S} = k_f(\mathbf{z}_*, \mathbf{z}_*; \boldsymbol{\theta}) - \mathbf{K}_{\mathbf{f}*}^T (\mathbf{K}_{\mathbf{f}} + \beta^{-1}\mathbf{I}_N)^{-1} \mathbf{K}_{\mathbf{f}*}$. Here $\mathbf{K}_{\mathbf{f}*}$ corresponds to cross-covariance at $\mathbf{Z}$ and $\mathbf{z}_*$.

The MCMC samples can be used to obtain an approximation to the marginalised predictive density in (3.17). However, the latent variable $\mathbf{z}_*$ cannot be marginalised analytically. Thus, given each sample of the chain $(\boldsymbol{\xi}^{(g)}, \mathbf{Z}^{(g)})$, we sample the latent variable $\mathbf{z}_*^{(g)}$ based on its predictive distribution in (3.18). The predictive density estimate is:

$$p\left(\mathbf{y}_* | \mathbf{x}_*, \mathbf{Y}, \mathbf{X}\right) \approx \frac{1}{G} \sum_{g=1}^{G} p\left(\mathbf{y}_* | \mathbf{z}_*^{(g)}, \mathbf{Z}^{(g)}, \mathbf{Y}, \boldsymbol{\theta}^{(g)}, \beta^{(g)}\right). \tag{3.19}$$

Similarly, the posterior mean function can be estimated by:

$$\mathbb{E}\left[\mathbf{y}_* | \mathbf{x}_*, \mathbf{Y}, \mathbf{X}\right] \approx \frac{1}{G} \sum_{g=1}^{G} \mathbf{K}_{\mathbf{f}*}^{(g)\,T} (\mathbf{K}_{\mathbf{f}}^{(g)} + \beta^{(g)\,-1}\mathbf{I}_N)\mathbf{Y}.$$

3.3.4 Example: Simulated sinusoidal data

In this section a comparison between the variational and pseudo-marginal inference frameworks is presented using a data set obtained from known trigonometric functions with artificially added noise. In simulating data this way each framework can be accurately compared to the truth. The data set is obtained by evaluating the data generating function:

$$f_{n,d}\left(\mathbf{x}_n\right) = \begin{cases} \zeta_d \cos(F_d \mathbf{x}_n) & \text{if } d = 1, 2, 3 \\ \zeta_d \sin(F_d \mathbf{x}_n) & \text{if } d = 4, 5, 6 \end{cases}, \tag{3.20}$$

at set of linearly spaced inputs between 0 and 4π, where $\mathbf{x}_n \in \mathbb{R}$ denotes the nth sample, and F_d is a factor of periodicity. Amplitudes are uniformly sampled from $\zeta_d \sim \mathcal{U}(0, 1)$ and kept consistent across examples, and the noise corrupted responses are obtained through $y_{n,d} = f_{n,d} + \varepsilon_{n,d}$, where $\varepsilon_{n,d} \overset{iid}{\sim} \mathcal{N}\left(0, 0.05^2\right)$. These functions are sampled under multiple parametrisations:

1. **Case 1** - A well-specified example where $F_d = 1 \forall d$.

2. **Case 2** - A poorly-specified example where $F_d \sim \mathcal{U}(0.8, 1.2) \forall d$, obtaining $F = (1.03, 0.92, 1.11, 0.99, 0.87, 1.02)$.

3. **Case 3** - A poorly-specified example where $F_d \sim \mathcal{U}(0.7, 1.3) \forall d$, obtaining $F = (1.04, 0.88, 1.15, 0.99, 0.80, 1.03)$.

In each case two latent dimensions are used with $N = 30$ samples. For comparison, the variational framework under two additional settings is also presented. For the first, the model is augmented with $k_z = 6$ latent dimensions (referred to as $\text{VEM}_{k_z=6}$) to demonstrate that two latent dimensions are sufficient for the first case[3], and that making the model well-specified in the second two cases does not change the outcome of the comparison. The second setting has $N = 60$ samples (this is referred to as $\text{VEM}_{N=60}$) to demonstrate that the advantages of PM persist even when optimisation is performed with a larger sample.

In the first case where the periods are constant it is expected that each inference scheme should be able to make adequate predictions. In the two additional cases where F_d is sampled from increasing uniform intervals, it is expected that point estimates of the hyperparameters will give an inadequate predictive distribution in the poorly specified cases. These examples are also designed to demonstrate the ability of PM to capture multi-modal posteriors. The improved uncertainty quantification and accuracy of predictions using PM is then demonstrated.

For both models a squared exponential kernel measures correlations in the input and latent spaces, with the addition of white noise (for numerical stability) on the preceding:

$$\begin{aligned} k_z\left(x, x'; \boldsymbol{\sigma}\right) =& \sigma_S \exp\left(-\frac{1}{2}\sigma_1 (x - x')^2\right) + \epsilon\delta\left(x, x'\right), \\ k_f(\mathbf{z}, \mathbf{z}'; \boldsymbol{\theta}) =& \theta_S \exp\left(-\frac{1}{2}\sum_{d=1}^{k_z} \theta_d (z_d - z_d')^2\right), \end{aligned}$$

where ϵ is a small positive constant and $\delta(\cdot, \cdot)$ is the kronecker-delta function. For identifiability the magnitude σ_S is fixed to one.

In all cases a Gamma prior is used on all hyperparameters, shown in Table 3.1, and a log transformation in the random walk proposals is used. The experiment was repeated for a number of prior parametrisations and it was found that predictive accuracy was not sensitive to prior choice. Four chains were run

[3]Through automatic relevance determination the model should prune unnecessary dimensions.

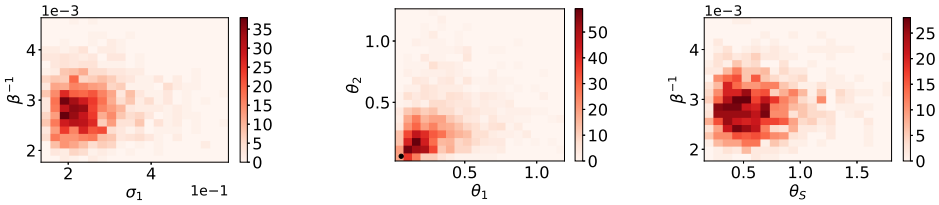
	σ_1	θ_1	θ_2	θ_S	β
Case 1	$\mathrm{Ga}\,(2, 8)$	$\mathrm{Ga}\,(1.25, 5)$	$\mathrm{Ga}\,(1.25, 5)$	$\mathrm{Ga}\,(1.5, 5)$	$\mathrm{Ga}\,(3, 800)$
Case 2	$\mathrm{Ga}\,(1.5, 16)$	$\mathrm{Ga}\,(2, 0.1)$	$\mathrm{Ga}\,(2, 0.1)$	$\mathrm{Ga}\,(2, 3)$	$\mathrm{Ga}\,(3, 800)$
Case 3	$\mathrm{Ga}\,(1.5, 16)$	$\mathrm{Ga}\,(2, 0.1)$	$\mathrm{Ga}\,(2, 0.1)$	$\mathrm{Ga}\,(2, 3)$	$\mathrm{Ga}\,(3, 800)$

Table 3.1: The hyper-prior distributions.

in parallel for 5000 iterations, adapting after $g_0 = 200$ iterations, and discarding the first 1000 as burn-in. Each chain was started at the approximate maximum marginal likelihood point-estimates with a small amount of noise. The collapsed blocked Gibbs sampler uses two blocks, $\boldsymbol{\xi}_1 = (\sigma_1, \theta_1, \theta_2)$ and $\boldsymbol{\xi}_2 = (\theta_S, \beta)$, using Algorithm 6. Rather than re-optimize the variational distribution after a full cycle of Gibbs iterations, it is re-optimised after each full conditional sample. Optimisation was performed until convergence, or until 1000 scaled conjugate gradient iterations had been performed.

Trace and autocorrelation plots demonstrate good mixing, and these plots for all cases are shown in figures A.1 and A.2. If necessary, mixing can be further improved by splitting the hyperparameters into smaller blocks. However, this would come at the price of an increased computational cost. The bivariate marginal posterior distribution for different pairs of hyperparameters is shown in Fig. 3.1, where the rows correspond to the three cases. Specifically, the pairs include the input lengthscale and model noise (σ_1, β^{-1}); latent lengthscales (θ_1, θ_2); and the signal variance and model noise (θ_S, β^{-1}). When the maximum marginal likelihood value lies within the axis it is marked with a dot. Note the tendency for the point-estimates of the variational approach to under-fit.

For the different cases, a comparison of the variational approximation to the true posterior is made using ESS. The conditional latent posterior distribution given the hyperparameters at state 820, $\theta^{(820)}$, of the collapsed Gibbs sampler is shown in Fig. A.3 and given the set of approximate maximum marginal likelihood hyperparameters, $\theta^{(ML)}$, obtained from jointly optimising over latent variables and hyperparameters is shown in Fig. A.4. These figures compare the quality of the variational approximation used in VEM to the true posterior used for predictions with the proposed PM inference scheme. Due to the high dimensional nature of these spaces, only bivariate contours, corresponding to two training samples, can be visualised at a time. In figures A.5 and A.6 the marginal conditional latent distribution for each sample is plotted alongside each other, given $\theta^{(820)}$ and $\theta^{(ML)}$ respectively. Also shown in figures A.7 to A.10 are the marginal distributions for the benchmark examples, given the approximate maximum marginal likelihood hyperparameters.



(a) Case 1: σ_1 v. β^{-1}. (b) Case 1: θ_1 v. θ_2. (c) Case 1: θ_S v. β^{-1}.

(d) Case 2: σ_1 v. β^{-1} (e) Case 2: θ_1 v. θ_2 (f) Case 2: θ_S v. β^{-1}

(g) Case 3: σ_1 v. β^{-1} (h) Case 3: θ_1 v. θ_2 (i) Case 3: θ_S v. β^{-1}

Figure 3.1: The hyperparameter joint posterior distributions for different pairs. The three rows correspond to the three data generating cases.

An inspection of these figures reveals that the true posterior of the latent variables is highly correlated and in many cases non-Gaussian, and the quality of variational approximation appears particularly poor with significant underestimation of the variance, especially as the model becomes increasingly miss-specified. It is noted that identifiability issues with latent variables may exaggerate the poor quality of the variational approximation.

The accuracy of the predictive densities under the variational and pseudo-marginal frameworks are compared in Table 3.2, in which the mean absolute error is reported. This is defined between samples of the true data generating function and predictive distribution of each framework. For an output d this is defined as:

$$\epsilon_d = \frac{1}{1000}\sum_{i=1}^{1000} |y^*_{i,d} - \tilde{y}^*_{i,d}|, \qquad y^*_{i,d} \sim \mathcal{N}\left(f_{i,d}\left(\mathbf{x}_i\right), 0.05^2\right), \tag{3.21}$$

where $\mathbf{x}_i$ are linearly spaced between 0 and 4π, and $\tilde{y}^*_{i,d}$ are samples of the predictive distributions in (3.15) and (3.19) at $\mathbf{x}_i$, for the variational and pseudo-marginal frameworks respectively. In addition to the three cases, errors for the VEM with an increased latent dimension, and with twice as many samples are also reported. For the first case increasing the latent dimension increased error. Given an automatic relevance determination kernel was used, additional dimensions should theoretically have been automatically pruned. Clearly effect did not occur, likely due to the increased number of variational parameters which needed to be optimised.

The predictive densities for the first feature of the third case, obtained using each inference scheme, are shown in Fig. 3.2, with a comparison to the true data generating function. In addition, the predictive densities for all features and cases are provided in the appendix figures A.11 to A.13.

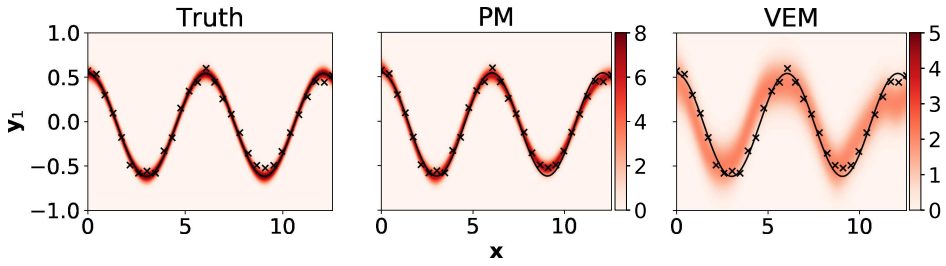


Figure 3.2: Case 3 predictive densities for the first feature. The mean of the data generating function is given as a solid line, while scatter points depict the training data.

The variational scheme clearly overestimates the uncertainty. We observe that PM gives a marked increase in accuracy across all features, particularly for the poorly specified examples. It must be noted that the VEM optimisation is over a non-convex function and therefore the maximum marginal likelihood found may not reflect the global optimum solution, despite convergence. This is particularly as the optima were very sensitive to initialisation, and moreover optimisation was over a lower bound to the marginal likelihood. However, this further necessitates posterior sampling in many cases.

3.4 Numerical computation

The asymptotic convergence guarantees of the pseudo-marginal scheme come at the cost of an additional computational burden, requiring repeated variational approximations to the marginal likelihood. This cost can be reduced by using stochastic

		y_1	y_2	y_3	y_4	y_5	y_6
Case 1	PM	**0.057**	**0.052**	**0.056**	**0.056**	**0.055**	**0.056**
	VEM	0.070	0.069	0.082	0.067	0.087	0.069
	$\text{VEM}_{k_z=6}$	0.115	0.093	0.110	0.106	0.126	0.105
	$\text{VEM}_{N=60}$	0.071	0.068	0.081	0.069	0.079	0.074
Case 2	PM	**0.067**	**0.057**	**0.054**	**0.057**	**0.053**	**0.056**
	VEM	0.111	0.131	0.168	0.112	0.165	0.128
	$\text{VEM}_{k_z=6}$	0.107	0.097	0.108	0.087	0.110	0.098
	$\text{VEM}_{N=60}$	0.077	0.080	0.092	0.074	0.088	0.075
Case 3	PM	**0.066**	**0.057**	**0.054**	**0.058**	**0.053**	**0.056**
	VEM	0.181	0.204	0.173	0.138	0.164	0.178
	$\text{VEM}_{k_z=6}$	0.101	0.088	0.107	0.082	0.095	0.089
	$\text{VEM}_{N=60}$	0.119	0.119	0.124	0.101	0.117	0.107

Table 3.2: The mean absolute error between samples of the true data generating distribution and the predictive distribution for each model and each case.

gradients, fewer optimiser iterations, more intelligent initialisation, or performing the variational approximation less frequently. However, for some examples these changes may also slow the convergence and mixing of the Markov Chain.

Alternatively we may also speed up our algorithm, introduced in section 3.3, using the GP-GIMH algorithm of Drovandi et al. [2018], in which a Gaussian process is used to approximate the marginal log likelihood. When the predictive variance is within a threshold, the Gaussian process can then be used to replace the variational approximation, avoiding an optimisation procedure. Whilst this sacrifices the asymptotic convergence guarantees of our algorithm, the approach will still benefit by avoiding the strong distributional assumptions of the variational framework.

Additionally, the MCMC scheme can be ran in parallel trivially, leading to a significant decrease in computational time. To scale to larger samples sizes, the proposed pseudo-marginal scheme can be combined with ideas from Hensman et al. [2015] and the approximate variational distribution used as a proposal in importance sampling can be replaced with the doubly stochastic variational scheme for deep Gaussian processes, recently proposed in Salimbeni and Deisenroth [2017]. However, it is noted that this comes at the cost of approximations to the sGPLVM in the pseudo-marginal framework, in order to scale to larger data sets.

3.5 Discussion

In models with strong correlations between parameters, Gibbs sampling is known to perform poorly (Titsias et al. [2009]). Strong correlations between variables

can result in inefficient mixing and slow convergence, and dependence in hierarchical models can lead to local behaviour of the tuning parameters which cannot be adapted without breaking detailed balance. Through the use of a pseudo-marginal scheme, the high correlations between latent variables and hyperparameters are broken. Simulated examples were presented which demonstrate the significant improvements that can be obtained through the pseudo-marginal inference scheme, particularly in the poorly-specified examples when point estimates of hyperparameters are insufficient.

By employing the KL divergence in the variational approximation, the latent variable posterior variance is underestimated. This does not affect the pseudo-marginal algorithm, which has Monte Carlo (MC) convergence guarantees. Of course, the closer the pseudo-marginal approximation is to the marginal, the faster the chain converges. Similarly the predictions are unaffected as latent variables are sampled using ESS, after taking advantage of pseudo-marginalisation to collapse the Gibbs sampler.

Although not observed in this chapter, variability in the pseudo-marginal estimates can induce 'stickiness' in the Markov Chain, in which randomly estimating a larger pseudo-marginal leads to a state from which it can be improbable to transition from. In this case, the variance of the pseudo-marginal estimates can be reduced using Pareto smoothed importance sampling (Vehtari et al. [2015]), or annealed importance sampling (Filippone [2013]). Alternatively, the pseudo-marginal can be re-estimated on each state transition.

In recent years deep learning has become a popular area of research. Many deep learning models, such as deep Gaussian processes, rely on variational approximations, both for scaling to large data sets and for analytic tractability. Although the methodology proposed here should readily extend to many such models, when the parameter space is of a higher dimension we would suggest the use of a pseudo HMC scheme on the collapsed probability model (Lindsten and Doucet [2016]).

Chapter 4

Uncertainty quantification with surrogate models

Monte Carlo (MC) sampling is the default method for investigating uncertainties in a system (e.g., propagating uncertainty in the inputs). MC estimates are extracted from multiple runs of the model using different realisations of the inputs, sampled from some distribution. While convergence is guaranteed as the number of runs increases, the slow rate of convergence demands (typically) a few thousand runs in order to extract reliable estimates of the statistics. If the model is computationally expensive, such a brute-force approach can be extremely time consuming or perhaps even infeasible (Maxwell et al. [2007]). Analytical stochastic methods have also been employed (Gelhar and Axness [1983] and Gelhar [1986]). Such methods can be useful for conceptual understanding of the process but are not applicable to practical scenarios.

Such limitations and shortcomings could be resolved in theory by using surrogate models (also known as metamodels, emulators or simply surrogates) in place of the complex numerical codes. That is, computationally-efficient approximations of the codes based on data-driven or reduced-order-model approaches. Another popular surrogate modelling approach is the stochastic-collocation method (Babuška et al. [2007]) in which the approximate response is constrained on a subspace, typically spanned by a generalised Polynomial Chaos basis (Xiu and Karniadakis [2002]). The coefficients in this basis are approximated *via* a collocation scheme. While these schemes yield good convergence rates, they scale poorly with the number of collocation points (Rajabi et al. [2015]). Although sparse grid methods based on the Smolyak algorithm (Smolyak [1963]) help to alleviate the increased computational burden, the resulting schemes are still severely limited by the input space dimen-

sionality and tend to perform poorly with limited observations (Xiu and Hesthaven [2005], Xiu [2007], Nobile et al. [2008] and Ma and Zabaras [2009]).

When data is scarce, we may turn to statistical Bayesian approaches such as Gaussian process regression. The first applications of GP surrogate models to uncertainty quantification can be found in O'Hagan and Kingman [1978]. See also the seminal papers of Currin et al. [1988] and Sacks et al. [1989]. GPs excel when data is scarce since they make *a priori* assumptions with regards to the relationship between data points. Comparatively, artificial neural networks (ANNs) make fewer *a priori* assumptions and as a result require much larger data sets; they are, therefore, infrequently used for uncertainty quantification tasks. In the context of groundwater flow, very few applications of GPs can be found (Bau and Mayer [2006], Hemker et al. [2008] and Borgonovo et al. [2012], the most likely explanations for which are the difficulty in implementing multi-output GP models and the lack of available information on, and software for GP modelling in comparison with ANNs. Existing applications again deal with low dimensional outputs, e.g., in Bau and Mayer [2006], the authors use a GP model to learn 4 well extraction rates for a pump-and-treat optimization problem.

There exist a number of challenges we may face when applying these machine learning methods to some data sets. For instance when data lies in a high dimensional space, inference becomes challenging. First, we face the curse of dimensionality, where our sample size must scale exponentially with the number of dimensions to avoid sparsity. Second, when using methods based on distance metrics, quantification of similarity between samples becomes more difficult Aggarwal et al. [2001]. This is most significant for a large number of covariates.[1] Often these high dimensional spaces have correlated covariates/features so we can overcome this problem using manifold learning to project points onto a lower dimensional space.

A novel, data driven framework for UQ in fluid dynamic models is introduced in this chapter. Particularly, the forward problem of uncertainty quantification is addressed, where variability in a parametrisation or input of a physical model (for example porosity) induces variability in the model response which must be quantified. It is often the case that dynamics of a fluid are modelled over a spatial domain, with each sample (corresponding to a unique input/parametrisation) containing values defined over a dense spatial mesh. Given the challenges of performing inference in a high dimensional space, this motivates a framework that incorporates manifold learning with emulation, allowing inference to be performed on latent projections

[1] Projecting points through any function cannot increase information entropy, and so the intrinsic dimensionality of the feature space cannot exceed that of the input space.

of model outputs. This manifold learning is performed using local tangent space alignment (LTSA), a nonparametric method that provides an automatic pre-image map. A diagram for the surrogate model framework is given in Fig 4.1.

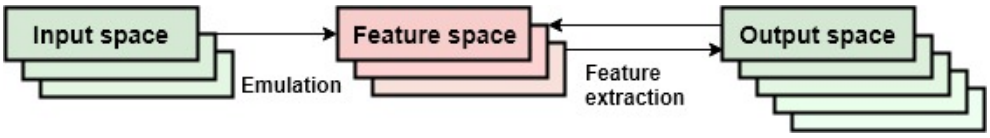


Figure 4.1: Framework diagram for the uncertainty quantification surrogate model

This chapter begins with an introduction to LTSA, which is used for the feature extraction step. Following this the Gaussian process model for emulation in the reduced feature space is presented, followed by a framework for combining these steps to perform uncertainty quantification. Finally, this framework is applied to two models for groundwater contamination.

4.1 Feature extraction for high dimensional spaces

For many examples, outputs in high dimensional spaces are correlated. Consequently, there exists an embedded linear manifold on which the samples lie. Manifold learning methods find this manifold and a mapping between these spaces. These can then be used for feature extraction, and inference can then be performed in a reduced feature space.

Definition 3. *A smooth k_z-manifold is defined as a topological space $\mathcal{Y}$ that is equipped with a maximal open cover $\{U_\alpha\}_{\alpha\in\Gamma}$ consisting of coordinate neighbourhoods (or patches) U_α, together with a collection of homeomorphisms (coordinate charts) $\phi_\alpha : U_\alpha \to \phi_\alpha(U_\alpha) \subset \mathbb{R}^{k_z}$ onto open subsets $\phi_\alpha(U_\alpha) \subset \mathbb{R}^{k_z}$ such that $\phi_\alpha(U_\alpha \cap U_\beta)$ and $\phi_\beta(U_\alpha \cap U_\beta)$ are open in $\mathbb{R}^{k_z}$; we say that ϕ_α and ϕ_β are compatible. Moreover, the transition maps defining a change of coordinates $\phi_\beta \circ \phi_\alpha^{-1}$ are diffeomorphisms for all $\alpha, \beta \in \Gamma$.*

Roughly speaking, a k_z-dimensional manifold $\mathcal{Y}$ is a set for which all points can be parametrised by k_z independent variables. A parametrisation is called a coordinate system (or a chart).

Let $\mathcal{A} = \{(U_\alpha, \phi_\alpha)\}_{\alpha\in\Gamma}$ be an atlas on $\mathcal{Y}$ ($\{U_\alpha\}_{\alpha\in\Gamma}$ is a cover and the $\{\phi_\alpha\}_{\alpha\in\Gamma}$ are pairwise compatible). Two smooth curves $\gamma_0, \gamma_1 : \mathbb{R} \to \mathcal{Y}$ are called $\mathbf{y}$-equivalent at a point $\mathbf{y} \in \mathcal{Y}$ if for every $\alpha \in \Gamma$ with $\mathbf{y} \in U_\alpha$, we have $\gamma_0(0) = \gamma_1(0) = \mathbf{y}$ and furthermore $(d/dt)|_{t=0}\phi_\alpha(\gamma_0(t)) = (d/dt)|_{t=0}\phi_\alpha(\gamma_1(t))$. With this equivalence relation, the equivalence class of a smooth curve γ with $\gamma(0) = v$ is denoted $[\gamma]_p$ and

the *tangent space* $T_{\mathbf{y}}\mathcal{Y}$ of $\mathcal{Y}$ at $\mathbf{y}$ is the set of equivalence classes $\{[\gamma]_p : \gamma(0) = \mathbf{y}\}$. The tangent space is a k_z-dimensional vector space, which is seen more clearly by identifying $T_{\mathbf{y}}\mathcal{Y}$ with the set of all derivations at $\mathbf{y}$ (linear maps from $C^\infty(\mathcal{Y})$ to $\mathbb{R}$ satisfying the derivation (Liebnitz) property).

4.1.1 Latent feature space representation

We assume that the output space $\mathcal{Y} \supset \mathbf{Y}$ is a manifold of dimension $k_z \ll k_y$ embedded in $\mathbb{R}^{k_y}$. Representations of points in $\mathcal{Y}$ and corresponding representations in the feature or latent space $\mathcal{F} \subset \mathbb{R}^{k_z}$ can be related by some smooth and *unknown* function $\mathbf{f} : \mathcal{F} \to \mathcal{Y}$. *Manifold learning* is concerned with the reconstruction of $\mathbf{f}$ and its inverse, given data points on the manifold. *Dimensionality reduction*, on the other hand, is concerned with the representation of given points in $\mathcal{Y}$ by corresponding points in the feature space $\mathcal{F}$. In this section we are interested primarily in dimensionality reduction and use *local tangent space alignment* (Zhang and Zha [2004]). The tangent space at a point $\mathbf{y}$ provides a low dimensional linear approximation of points in a neighbourhood of $\mathbf{y}$. We can approximate each point $\mathbf{y}$ in a data set using a basis for $T_{\mathbf{y}}\mathcal{Y}$ and use these approximations to find low dimensional representations in a global coordinate system, by aligning the tangent spaces using local affine transformations (Zhang and Zha [2004]). We note that this assumes the existence of a single chart (homeomorphism) $\mathbf{f}^{-1}$.

Consider a noise-free model in which the data $\mathbf{Y}$ is generated by the smooth function $\mathbf{f}$ defined above:

$$\mathbf{y} = \mathbf{f}(\mathbf{z}) = \big(f_1(\mathbf{z}), \ldots, f_{k_y}(\mathbf{z})\big)^T, \tag{4.1}$$

where $\mathbf{z} = (z_1, \ldots, z_{k_z})^T \in \mathcal{F}$ is a latent feature vector (i.e., the low dimensional representation of the point $\mathbf{y}$). Under the assumption that $\mathbf{f}$ is smooth, it can be approximated using a first-order Taylor expansion in a neighbourhood $\Omega(\mathbf{z})$ of a point $\mathbf{z}$: $\mathbf{f}(\widehat{\mathbf{z}}) = \mathbf{f}(\mathbf{z}) + \mathbf{J}_{\mathbf{f}}(\mathbf{z}) \cdot (\widehat{\mathbf{z}} - \mathbf{z}) + \mathcal{O}(\|\widehat{\mathbf{z}} - \mathbf{z}\|^2)$, $\forall \widehat{\mathbf{z}} \in \Omega(\mathbf{z})$, where $\mathbf{J}_{\mathbf{f}}(\mathbf{z}) \in \mathbb{R}^{k_y \times k_z}$ is the Jacobi matrix of $\mathbf{f}$ at $\mathbf{z}$, the i,j-th entry of which is $\partial f_i / \partial z_j$. Here and throughout, $||\cdot||$ denotes a standard Euclidean norm.

A basis for the tangent space $T_{\mathbf{y}}\mathcal{Y}$ of $\mathcal{Y}$ (a k_z-dimensional linear subspace of $\mathbb{R}^{k_y}$) at $\mathbf{y} = \mathbf{f}(\mathbf{z})$ is given by the span of the column vectors of $\mathbf{J}_{\mathbf{f}}$. The vector $\widehat{\mathbf{z}} - \mathbf{z}$ then gives the coordinate of $\mathbf{f}(\widehat{\mathbf{z}})$ in the affine subspace $\mathbf{f}(\mathbf{z}) + T_{\mathbf{y}}\mathcal{Y}$. $\mathbf{J}_{\mathbf{f}}$ cannot be computed explicitly without knowledge of $\mathbf{f}$. Suppose we can express $T_{\mathbf{y}}\mathcal{Y}$ in terms

of a matrix $\mathbf{Q_z}$, the columns of which form an orthonormal basis for $T_{\mathbf{y}}\mathcal{Y}$:

$$\mathbf{J}_f(\mathbf{z}) \cdot (\widehat{\mathbf{z}} - \mathbf{z}) = \mathbf{Q_z}\boldsymbol{\pi}^*_{\mathbf{z}}, \tag{4.2}$$

where $\boldsymbol{\pi}^*_{\mathbf{z}} = \mathbf{Q}^T_{\mathbf{z}}\mathbf{J}_{\mathbf{f}}(\mathbf{z}) \cdot (\widehat{\mathbf{z}} - \mathbf{z}) \equiv \mathbf{P_z}\,(\widehat{\mathbf{z}} - \mathbf{z})$ is still unknown. Combining (4.2) with the Taylor expansion, we can, however, find an approximation of $\boldsymbol{\pi}^*_{\mathbf{z}}$ consisting of an orthogonal projection of $\mathbf{f}\,(\widehat{\mathbf{z}}) - \mathbf{f}(\mathbf{z})$ onto $T_{\mathbf{y}}\mathcal{Y}$:

$$\boldsymbol{\pi}_{\mathbf{z}} \equiv \mathbf{Q}^T_{\mathbf{z}}\left(\mathbf{f}\,(\widehat{\mathbf{z}}) - \mathbf{f}(\mathbf{z})\right) = \boldsymbol{\pi}^*_{\mathbf{z}} + \mathcal{O}\left(\|\widehat{\mathbf{z}} - \mathbf{z}\|^2\right), \tag{4.3}$$

provided that the basis $\mathbf{Q_z}$ is known for each $\mathbf{z}$. Truncating this expansion, the global coordinate $\mathbf{z}$ then satisfies:

$$\int\int_{\Omega(\mathbf{z})} \|\mathbf{P_z}\,(\widehat{\mathbf{z}} - \mathbf{z}) - \boldsymbol{\pi}_{\mathbf{z}}\| d\widehat{\mathbf{z}} \approx 0. \tag{4.4}$$

If the Jacobian is of full column rank we can find a local affine transformation:

$$\widehat{\mathbf{z}} - \mathbf{z} \approx \mathbf{P}^{-1}_{\mathbf{z}}\boldsymbol{\pi}_{\mathbf{z}} \equiv \mathbf{L_z}\boldsymbol{\pi}_{\mathbf{z}}. \tag{4.5}$$

The transformation $\mathbf{L_z}$ aligns the local coordinate with the global coordinate $\widehat{\mathbf{z}} - \mathbf{z}$ for $f(\widehat{\mathbf{z}})$. We then find the global coordinate $\widehat{\mathbf{z}}$ and affine transformation $\mathbf{L_z}$ by minimizing $\int\int_{\Omega(\mathbf{z})} \|\widehat{\mathbf{z}} - \mathbf{z} - \mathbf{L_z}\boldsymbol{\pi}_{\mathbf{z}}\| d\widehat{\mathbf{z}}$.

We note that the orthogonal basis $\mathbf{Q_z}$ for each tangent space is still unknown. Consider a data set $\mathbf{y}_n$, $n = 1, \ldots, N$, sampled with noise ϵ_n, $n = 1, \ldots, N$, from the underlying non-linear manifold:

$$\mathbf{y}_n = \mathbf{f}(\mathbf{z}_n) + \epsilon_n. \tag{4.6}$$

For any $\mathbf{y}_n$, let $\mathbf{Y}_n = [\mathbf{y}_{n_1} \ldots \mathbf{y}_{n_P}]$ be the matrix containing the P nearest neighbours, including $\mathbf{y}_n$, where distances are measured using the standard Euclidean metric. The best k_z-dimensional local affine subspace approximation for the points in $\mathbf{Y}_n$ is given by:

$$\arg\min_{\mathbf{y},\boldsymbol{\Pi},\mathbf{Q}} \sum_{k=1}^{P} \|\mathbf{y}_{n_k} - (\mathbf{y} + \mathbf{Q}\boldsymbol{\pi}_k)\|_2^2 = \arg\min_{\mathbf{y},\boldsymbol{\Pi},\mathbf{Q}} \|\mathbf{Y}_n - \left(\mathbf{y}\mathbf{e}^T + \mathbf{Q}\boldsymbol{\Pi}\right)\|_2^2, \tag{4.7}$$

in which the orthonormal matrix $\mathbf{Q}$ has k_z columns, $\boldsymbol{\Pi} = [\boldsymbol{\pi}_1 \ldots \boldsymbol{\pi}_P]$ and $\mathbf{e}$ is a vector of all ones. The optimal $\mathbf{y}$ is given by the mean of $\{\mathbf{y}_{n_k}\}_k$, denoted $\bar{\mathbf{y}}_n$, and the optimal $\mathbf{Q}$ is given by $\mathbf{Q}_n$, the columns of which are the k_z left singular vectors of

$\mathbf{Y}_n\left(\mathbf{I}-\mathbf{e}\mathbf{e}^T/P\right)$ corresponding to the k_z largest singular values. Lastly, $\mathbf{\Pi}$ is given by $\mathbf{\Pi}_n$:

$$\mathbf{\Pi}_n = \mathbf{Q}_n^T\mathbf{Y}_n\left(\mathbf{I}-\frac{1}{P}\mathbf{e}\mathbf{e}^T\right) = \left[\boldsymbol{\pi}_1^{(i)},\ldots,\boldsymbol{\pi}_K^{(i)}\right], \tag{4.8}$$

where $\boldsymbol{\pi}_k^{(i)} = \mathbf{Q}_n^T\left(\mathbf{y}_{n_k}-\bar{\mathbf{y}}_n\right)$. Consequently:

$$\mathbf{y}_{n_k} = \bar{\mathbf{y}}_n + \mathbf{Q}_n\boldsymbol{\pi}_k^{(l)} + \varphi_k^{(l)}, \tag{4.9}$$

where $\varphi_k^{(l)} = \left(I-\mathbf{Q}_n\mathbf{Q}_n^T\right)\left(\mathbf{y}_{n_k}-\bar{\mathbf{y}}_n\right)$ is the reconstruction error. Having minimised the local reconstruction error, we would like to find the global coordinates $\mathbf{Z} = [\mathbf{z}_1\ldots\mathbf{z}_N] \in \mathbb{R}^{k_z\times N}$, corresponding to data points $\mathbf{Y}$, given the local coordinates $\boldsymbol{\pi}_k^{(l)}$. The global coordinates $\mathbf{z}_{n_k}$ of the corresponding points $\mathbf{y}_{n_k}$ are chosen to respect the local geometry as determined by the $\boldsymbol{\pi}_k^{(l)}$:

$$\begin{aligned}\mathbf{z}_{n_k} &= \bar{\mathbf{z}}_n + \mathbf{L}_n\boldsymbol{\pi}_k^{(l)} + \epsilon_k^{(l)}, \quad k=1,\ldots,P,\ l=1,\ldots,N,\\ \mathbf{Z}_n &= \frac{1}{P}\mathbf{Z}_n\mathbf{e}\mathbf{e}^T + \mathbf{L}_n\mathbf{\Pi}_n + \mathbf{E}_n,\end{aligned} \tag{4.10}$$

where $\bar{\mathbf{z}}_n$ is the mean of $\{\mathbf{z}_{n_k}\}_k$, $\mathbf{Z}_n = [\mathbf{z}_{n_1}\ldots\mathbf{z}_{n_P}]$ and $\mathbf{E}_n = [\epsilon_1^{(l)}\ldots\epsilon_P^{(l)}]$, given by $\mathbf{E}_n = \mathbf{Z}_n(\mathbf{I}-\mathbf{e}\mathbf{e}^T/P) - \mathbf{L}_n\mathbf{\Pi}_n$. We find the latent points and local affine transformations $\mathbf{L}_n$ that minimize the local reconstruction error $\|\mathbf{E}_n\|_F$, in which $||\cdot||_F$ denotes a Frobenius norm. The optimal $\mathbf{L}_n$ are given by $\mathbf{L}_n = \mathbf{Z}_n(\mathbf{I}-\mathbf{e}\mathbf{e}^T/P)\mathbf{\Pi}_n^+$, and consequently the errors are given by $\mathbf{E}_n = \mathbf{Z}_n(\mathbf{I}-\mathbf{e}\mathbf{e}^T/P)(\mathbf{I}-\mathbf{\Pi}_n^+\mathbf{\Pi}_n)$, where $\mathbf{\Pi}_n^+$ is the Moor-Penrose pseudo inverse of $\mathbf{\Pi}_n$. We define a 0-1 selection matrix $\mathbf{S}_n \in \mathbb{R}^{N\times P}$ such that $\mathbf{Z}\mathbf{S}_n = \mathbf{Z}_n$. The global coordinates can then be selected according to a minimization of the overall reconstruction error:

$$\underset{\mathbf{Z}:\mathbf{Z}^T\mathbf{Z}=\mathbf{I}}{\arg\min}\sum_n \|\mathbf{E}_n\|_F^2 = \underset{\mathbf{Z}:\mathbf{Z}^T\mathbf{Z}=\mathbf{I}}{\arg\min}\|\mathbf{Z}\mathbf{S}\mathbf{W}\|_F^2, \tag{4.11}$$

where $\mathbf{S} = [\mathbf{S}_1\ldots\mathbf{S}_N]$, and $\mathbf{W} = \mathrm{diag}\left(\mathbf{W}_1,\ldots,\mathbf{W}_N\right)$, in which $\mathbf{W}_n = (\mathbf{I}-\mathbf{e}\mathbf{e}^T/P)(\mathbf{I}-\mathbf{\Pi}_n^+\mathbf{\Pi}_n)$. The constraint $\mathbf{Z}^T\mathbf{Z} = \mathbf{I}$ ensures that the solutions are unique. The vector $\mathbf{e}$ is an eigenvector of $\mathbf{B} \equiv \mathbf{S}\mathbf{W}\mathbf{W}^T\mathbf{S}^T \in \mathbb{R}^{N\times N}$ corresponding to a zero eigenvalue. Arranging the eigenvalues in increasing order, the optimal $\mathbf{Z}$ is given by $\mathbf{Z}' = [\boldsymbol{\zeta}_2\ldots\boldsymbol{\zeta}_{k_z+1}]^T$, where $\boldsymbol{\zeta}_2,\ldots,\boldsymbol{\zeta}_{k_z+1} \in \mathbb{R}^N$ are the eigenvectors of $\mathbf{B}$ corresponding to the $(k_z+1)^{\text{st}}$ smallest eigenvalues excluding the first (zero) eigenvalue. This defines a map $\mathbf{f}^-:\mathbf{y}\mapsto\mathbf{z}$, $\mathbf{z}=\mathbf{f}^-(\mathbf{y})$, that approximates $\mathbf{f}^{-1}:\mathcal{Y}\to\mathcal{F}$ *for the given data points*:

$$\mathbf{z}_n = \mathbf{f}^{-1}(\mathbf{y}_n) \approx \mathbf{f}^-(\mathbf{y}_n) = \mathbf{z}'_{n,:}. \tag{4.12}$$

in which $\mathbf{z}'_{n,:}$ is the n-th column of $\mathbf{Z}'$.

Fixing the number of neighbours assumes that the manifold has a certain smoothness, while using the same number of neighbours for every tangent space assumes a global smoothness. These assumptions may result in inaccurate predictions, in which case we can use adaptive algorithms (Zou and Zhu [2011], Zhang et al. [2012] and Wei et al. [2008]). Similar adaptations can be made for other issues, such as robustness in the presence of noise (Zhan and Yin [2011]).

We remark that LTSA is a nonparametric technique, in that an explicit form of $\mathbf{f}$ is not available. This means that the *out-of-sample* problem does not have a parametric (explicit) solution. In other words, application of LTSA (the map $\mathbf{f}^-$) to a point that was not in the data set can only be achieved by re-running the entire algorithm with an updated data set that appends the new point. Non-parametric solutions to the out-of-sample problem have been developed, and one that is applicable to LTSA can be found in Li et al. [2005].

If we map points $\mathbf{y} \in \mathcal{Y}$ to $\mathcal{F}$ using $\mathbf{f}^-$ and perform inference in $\mathcal{F}$, an approximation of $\mathbf{f}$ is required in order to make predictions in the physical space $\mathcal{Y}$. This is referred to as the *pre-image* problem in manifold learning methods: given a point in the low dimensional space, find a mapping to the original space (manifold). We outline an approximation of the pre-image map in the next section.

4.1.2 Pre-image problem: Reconstructing outputs

Given a point $\mathbf{z} \in \mathcal{F}$ in latent space we require the corresponding point in the original physical space $\mathbf{y} \in \mathcal{Y}$. Let $\mathbf{z}_k$ be the neighbour nearest to $\mathbf{z}$. According to (4.10):

$$\boldsymbol{\pi}_*^{(k)} = \mathbf{L}_k^{-1}\left(\mathbf{z} - \bar{\mathbf{z}}_k\right) - \mathbf{L}_k^{-1}\epsilon_*^{(k)}. \tag{4.13}$$

By (4.9) we can also define:

$$\mathbf{y} = \bar{\mathbf{y}}_k + \mathbf{Q}_k\boldsymbol{\pi}_*^{(k)} + \varphi_*^{(k)}. \tag{4.14}$$

Consequently, we have the following approximate pre-image mapping $\hat{\mathbf{f}} : \mathcal{F} \to \mathcal{Y}$ (approximation of $\mathbf{f}$):

$$\begin{aligned}\mathbf{y} = \mathbf{f}(\mathbf{z}) \approx \hat{\mathbf{f}}(\mathbf{z}) &= \bar{\mathbf{y}}_k + \mathbf{Q}_k\left(\mathbf{L}_k^{-1}\left(\mathbf{z} - \bar{\mathbf{z}}_k\right) - \mathbf{L}_k^{-1}\epsilon_*^{(k)}\right) + \varphi_*^{(k)} \\ &= \bar{\mathbf{y}}_k + \mathbf{Q}_k\mathbf{L}_k^{-1}\left(\mathbf{z} - \bar{\mathbf{z}}_k\right) + \mathcal{E},\end{aligned} \tag{4.15}$$

where $k = \arg\min_n \|\mathbf{z} - \mathbf{z}_n\|$ and $\mathcal{E} = -\mathbf{Q}_k\mathbf{L}_k^{-1}\epsilon_*^{(k)} + \varphi_*^{(k)}$ incorporates the error terms.

4.2 Gaussian process emulation

The surrogate model problem is defined as one of approximating the simulator mapping $\boldsymbol{\eta} : \mathcal{X} \to \mathcal{Y}$ given the data set $\mathcal{D}' = \{\boldsymbol{\Xi}, \mathbf{Y}\}$ derived from runs of the simulator at selected design points $\{\boldsymbol{\xi}_n\}_{n=1}^N$. We can instead consider the simulator as a mapping $\boldsymbol{\eta}_{\mathcal{F}} \equiv \mathbf{f}^{-1} \circ \boldsymbol{\eta} : \mathcal{X} \to \mathcal{F}$ from the input space to the latent feature space, i.e., $\boldsymbol{\eta}_{\mathcal{F}}(\cdot) = \mathbf{f}^{-1}(\boldsymbol{\eta}(\cdot))$. Application of LTSA to points on the manifold approximates this mapping $\mathbf{f}^{-} \approx \mathbf{f}^{-1}$. The original data set $\mathcal{D}' = \{\boldsymbol{\Xi}, \mathbf{Y}\}$ is then replaced by the equivalent data set $\mathcal{D} = \{\boldsymbol{\Xi}, \mathbf{Z}\}$ or $\mathcal{D} = \{(\boldsymbol{\xi}_n, \mathbf{z}_n)\}_{n=1}^N$, where $\mathbf{z}_n = \mathbf{f}^{-}(\mathbf{y}_n) \approx \mathbf{f}^{-1}(\mathbf{y}_n) = \mathbf{f}^{-1}(\boldsymbol{\eta}(\boldsymbol{\xi}_n)) = \boldsymbol{\eta}_{\mathcal{F}}(\boldsymbol{\xi}_n)$, and our aim is now to approximate the mapping $\boldsymbol{\eta}_{\mathcal{F}}(\cdot)$. Returning a general point $\mathbf{z} = \boldsymbol{\eta}_{\mathcal{F}}(\boldsymbol{\xi})$ to the corresponding point $\mathbf{y}$ in the space $\mathcal{Y}$ is discussed in the next section.

A GP model is used to infer the mapping $\boldsymbol{\eta}_{\mathcal{F}} : \boldsymbol{\xi} \mapsto \mathbf{z}$ by treating it as a realization of a (Gaussian) stochastic process indexed by the inputs $\boldsymbol{\xi}$. Specifically, we learn the functional relationship for each latent feature of $\mathbf{z}$ separately (assuming independence) using a *scalar* GP model. Let $z_{n,i}$, $i = 1, \ldots, k_z$, denote the i-th component of $\mathbf{z}_n$, $n = 1, \ldots, N$. The probabilistic model is $z_{n,i} = h_i(\boldsymbol{\xi}_n) + \eta_{n,i}$, in which the signal noise distribution is $p(\eta_{n,i}) = \mathrm{N}(0, \beta_i^{-1})$ $\forall n$, where β_i is the precision. We assume independent zero-mean GP priors $h_i(\boldsymbol{\xi}) \sim \mathrm{GP}(0, c_h(\boldsymbol{\xi}, \boldsymbol{\xi}'; \boldsymbol{\theta}_i))$, where $c_h(\boldsymbol{\xi}, \boldsymbol{\xi}'; \boldsymbol{\theta}_i)$ is the kernel function (of the same form across i) in which $\boldsymbol{\theta}_i$ is a vector of hyperparameters pertaining to component i. The noise-free latent functions $h_i(\boldsymbol{\xi})$, $i = 1, \ldots, k_z$, can be thought of as independent draws from the GP. Using the notation $h_{n,i} = h_i(\boldsymbol{\xi}_n)$ we can define a matrix $\mathbf{H} \in \mathbb{R}^{N \times k_z}$ with columns $\mathbf{h}_{:,i} = (h_{1,i}, \ldots, h_{N,i})^T$. By the independence assumption:

$$p(\mathbf{H}|\boldsymbol{\Xi}, \boldsymbol{\Theta}) = \prod_{i=1}^{k_z} p(\mathbf{h}_{:,i}|\boldsymbol{\Xi}, \boldsymbol{\theta}_i), \tag{4.16}$$

where $\boldsymbol{\Theta} = [\boldsymbol{\theta}_1 \ldots \boldsymbol{\theta}_{k_z}]$. By the properties of GPs, we have $p(\mathbf{h}_{:,i}|\boldsymbol{\Xi}, \boldsymbol{\theta}_i) = \mathrm{N}(\mathbf{0}, \mathbf{C}_i)$, in which $\mathbf{C}_i \in \mathbf{R}^{N \times N}$ is a kernel matrix, the n, m-th entry of which is $c_h(\boldsymbol{\xi}_n, \boldsymbol{\xi}_m; \boldsymbol{\theta}_i)$. Thus:

$$\begin{aligned} p(\mathbf{Z}|\boldsymbol{\Xi}, \boldsymbol{\Theta}, \boldsymbol{\beta}) &= \int \prod_{i=1}^{k_z} \prod_{n=1}^{N} p(z_{n,i}|h_{n,i}, \beta_i) p(\mathbf{h}_{:,i}|\boldsymbol{\Xi}, \boldsymbol{\theta}_i) d\mathbf{H} \\ &= \prod_{i=1}^{k_z} \mathrm{N}(\mathbf{0}, \mathbf{C}_i + \beta_i^{-1}\mathbf{I}), \end{aligned} \tag{4.17}$$

in which $p(\mathbf{z}_{:,i}|\mathbf{h}_{:,i}, \beta_i) = \mathrm{N}(\mathbf{h}_{:,i}, \beta_i^{-1}\mathbf{I})$ by virtue of the noise model, and $\boldsymbol{\beta} =$

$(\beta_1, \ldots, \beta_{k_z})^T$.

We place gamma priors on all hyperparameters $\boldsymbol{\theta}_i$ and signal noise (likelihood nugget) precisions β_i. The parametrisation of these priors is determined through an initial MML optimiser run. We then choose these parameters such that the mean is equal to the MML estimate, and so that we obtain an appropriate variance. We sample our hyperparameter posterior using HMC, outlined in section 2.5.2. Let $\mathbf{z} \in \mathcal{F}$ be the feature vector corresponding to the test (new) input $\boldsymbol{\xi}$. The predictive distribution for the i^{th} component z_i of $\mathbf{z}$ $(i = 1, \ldots, k_z)$ is given by:

$$\begin{aligned} p\left(z_i|\boldsymbol{\xi}, \mathcal{D}, \boldsymbol{\theta}_i, \beta_i\right) &= \mathcal{N}\left(\mu_i(\boldsymbol{\xi}), \sigma_i^2(\boldsymbol{\xi})\right), \\ \mu_i(\boldsymbol{\xi}) &= \mathbf{c}_h(\boldsymbol{\xi}, \boldsymbol{\Xi}; \boldsymbol{\theta}_i)^T \left(\mathbf{C}_i + \beta_i^{-1}\mathbf{I}\right)^{-1} \mathbf{z}_{:,i}, \\ \sigma_i^2(\boldsymbol{\xi}) &= c_h\left(\boldsymbol{\xi}, \boldsymbol{\xi}; \boldsymbol{\theta}_i\right) - \mathbf{c}_h(\boldsymbol{\xi}, \boldsymbol{\Xi}; \boldsymbol{\theta}_i)^T \left(\mathbf{C}_i + \beta_i^{-1}\mathbf{I}\right)^{-1} \mathbf{c}_h(\boldsymbol{\xi}, \boldsymbol{\Xi}; \boldsymbol{\theta}_i), \end{aligned} \tag{4.18}$$

where $\mathbf{c}_h(\boldsymbol{\xi}, \boldsymbol{\Xi}; \boldsymbol{\theta}_i) = (c_h(\boldsymbol{\xi}_1, \boldsymbol{\xi}; \boldsymbol{\theta}_i), \ldots, c_h(\boldsymbol{\xi}_N, \boldsymbol{\xi}; \boldsymbol{\theta}_i))^T \in \mathbb{R}^N$ is the cross covariance between $\mathbf{z}$ and $\mathbf{z}_n$, $n = 1, \ldots, N$. Thus, the latent variable GP prediction is distributed as:

$$\begin{aligned} p(\mathbf{z}|\boldsymbol{\xi}, \mathcal{D}, \boldsymbol{\Theta}, \boldsymbol{\beta}) &= \mathrm{N}(\boldsymbol{\mu}_{\mathbf{z}}(\boldsymbol{\xi}), \Sigma_{\mathbf{z}}(\boldsymbol{\xi})), \\ \boldsymbol{\mu}_{\mathbf{z}}(\boldsymbol{\xi}) &= (\mu_1(\boldsymbol{\xi}), \ldots, \mu_{k_z}(\boldsymbol{\xi}))^T, \\ \Sigma_{\mathbf{z}}(\boldsymbol{\xi}) &= \mathrm{diag}(\sigma_1^2(\boldsymbol{\xi}), \ldots, \sigma_{k_z}^2(\boldsymbol{\xi})), \end{aligned} \tag{4.19}$$

where the components of $\boldsymbol{\mu}_{\mathbf{z}}(\boldsymbol{\xi}) \in \mathcal{F}$ are given by the second line of (4.18) and $\Sigma_{\mathbf{z}}(\boldsymbol{\xi}) \in \mathbb{R}^{k_z \times k_z}$ is a diagonal covariance matrix, in which the i-th diagonal element corresponds to the predictive variance given by the third line of (4.18), while the off diagonal elements are zero due to the assumption of independent GPs across i.

4.3 Predictions

The physical models we consider have an unknown, stochastic input (e.g., the hydraulic conductivity). This represents a lack of knowledge of the input, which induces a random variable response (e.g., the pressure head). Quantifying the distribution over the response is referred to as a pushforward or *forward problem*. The *pushforward measure* is the distribution over the response, or quantity of interest derived from the response[2]. Based on the methods of the preceding sections, we

[2]Let $\mathbb{P}_{\mathcal{X}}$ be a measure on $(\mathcal{X}, \mathcal{F}_{\mathcal{X}})$. The pushforward measure of $\mathbb{P}_{\mathcal{X}}$ under $\boldsymbol{\eta} : (\mathcal{X}, \mathcal{F}_{\mathcal{X}}, \mathbb{P}_{\mathcal{X}}) \to (\mathcal{Y}, \mathcal{F}_{\mathcal{Y}}, \mathbb{P}_{\mathcal{Y}})$ is defined as $\mathbb{P}_{\mathcal{Y}}(F) = \mathbb{P}_{\mathcal{X}} \circ \boldsymbol{\eta}^{-1}(F)$ for $F \in \mathcal{F}_{\mathcal{Y}}$. We characterize the measures by their probability density functions (pdfs) with respect to Lebesgue measure. In this work a Gaussian distribution is placed on the inputs.

now present an emulation framework for interrogating the pushforward measure (the response distribution). We begin by describing in the next section how a single realization of the random variable response may be obtained given a single realization of the stochastic input. In section 4.3.2, we then discuss how to quantify the pushforward measure (extract relevant statistics of the response).

4.3.1 Conditional predictions

Due to the nature of the emulator, the prediction of a point $\mathbf{z} \in \mathcal{F}$ is normally distributed. This distribution captures uncertainty in the predictions as a consequence of limited and noise corrupted data. A common challenge when using reduced dimensional representations is analytically propagating this distribution through a non-linear, pre-image map (in this case $\hat{\mathbf{f}} : \mathcal{F} \ni \mathbf{z} \mapsto \mathbf{y} \in \mathcal{Y}$ defined by (4.15)) for a test input $\boldsymbol{\xi}$.

Analytically propagating a distribution through a non-linear mapping is often not feasible. Instead we could repeatedly sample from the feature-space response distribution (over $\mathbf{z} \in \mathcal{F}$) and apply the pre-image map to find the distribution over the corresponding $\mathbf{y} \in \mathcal{Y}$. Examples that use this latter approach include kernel principal component analysis and Gaussian process latent variable models. In the latter case, approximations can be obtained using the projected process approximation. Since the manifold consists of aligned (tangent) hyperplanes, however, we are able to derive locally linear pre-image maps that can be used for mapping distributions defined on local tangent spaces. The latent variable GP prediction $\mathbf{z}$ is distributed according to (4.19). Restricting to a single tangent space, it is a straightforward task to push this distribution through (4.15) to obtain a normal distribution for the corresponding $\mathbf{y} \in \mathcal{Y}$:

$$\begin{aligned} p(\mathbf{y}|\boldsymbol{\xi}, \mathcal{D}, \boldsymbol{\Theta}, \boldsymbol{\beta}) &= \mathcal{N}\left(\boldsymbol{\mu}_{\mathbf{y}}(\boldsymbol{\xi}), \Sigma_{\mathbf{y}}(\boldsymbol{\xi})\right), \\ \boldsymbol{\mu}_{\mathbf{y}}(\boldsymbol{\xi}) &= \bar{\mathbf{y}}_k + \mathbf{Q}_k \mathbf{L}_k^{-1} \left(\boldsymbol{\mu}_{\mathbf{z}}(\boldsymbol{\xi}) - \bar{\mathbf{z}}_k\right), \\ \Sigma_{\mathbf{y}}(\boldsymbol{\xi}) &= \mathbf{Q}_k \mathbf{L}_k^{-1} \Sigma_{\mathbf{z}}(\boldsymbol{\xi}) \left(\mathbf{Q}_k \mathbf{L}_k^{-1}\right)^T, \end{aligned} \tag{4.20}$$

where $k = \arg\min_n \|\boldsymbol{\mu}_{\mathbf{z}}(\boldsymbol{\xi}) - \mathbf{z}_n\|$, $\boldsymbol{\mu}_{\mathbf{y}}(\boldsymbol{\xi}) \in \mathbb{R}^{k_y}$, and $\Sigma_{\mathbf{y}}(\boldsymbol{\xi}) \in \mathbb{R}^{k_y \times k_y}$. This result is particularly useful for scenarios in which knowledge of the correlations between outputs is required. Without this result we would require a large number of samples to estimate the covariance (tens of thousands). If, however, we are only interested in samples of the distribution (4.19), i.e, making predictions at specified inputs, then it is more memory efficient to sample from the predictive distribution (4.19) and use the pre-image map (4.15). When the support of this distribution over latent

features is large, the accuracy of the local approximation breaks down so we must first sample the latent features before applying the pre-image map.

4.3.2 Predictions marginalizing the stochastic input

Having obtained a distribution over the response for a stochastic input realization, we now consider the problem of obtaining a distribution over the response marginalised over the stochastic input. We assume that the input is normally distributed:

$$p(\boldsymbol{\xi}) = \mathcal{N}\left(\boldsymbol{\mu}_{\boldsymbol{\xi}}, \Sigma_{\boldsymbol{\xi}}\right), \tag{4.21}$$

for some mean vector $\boldsymbol{\mu}_{\boldsymbol{\xi}}$ (equal to $\mathbf{0}$ in this case) and covariance matrix $\Sigma_{\boldsymbol{\xi}}$ (equal to $\mathbf{I}$ in this case). We wish to evaluate:

$$p\left(\mathbf{y}|\mathcal{D}, \boldsymbol{\Theta}, \boldsymbol{\beta}\right) = \hat{\mathbf{f}}\left(p(\mathbf{z}|\mathcal{D}, \boldsymbol{\Theta}, \boldsymbol{\beta})\right) = \hat{\mathbf{f}}\left(\int p(\mathbf{z}|\boldsymbol{\xi}', \mathcal{D}, \boldsymbol{\Theta}, \boldsymbol{\beta}) p(\boldsymbol{\xi}') d\boldsymbol{\xi}'\right), \tag{4.22}$$

where $\hat{\mathbf{f}}$ is the (measurable) pre-image map and $p\left(\mathbf{z}|\boldsymbol{\xi}, \mathcal{D}, \boldsymbol{\Theta}, \boldsymbol{\beta}\right)$ is defined in (4.19). Since the input $\boldsymbol{\xi}$ appears non-linearly in the inverse of the $\mathbf{z}$ predictive distribution covariance $\sigma^2(\boldsymbol{\xi})$, we are unable to find a closed form solution to the integral in (4.22), i.e., the marginal distribution over $\mathbf{z}$. The moments of this marginal distribution can, on the other hand, be found analytically, although we will not know the family of distributions to which these moments belong.

Let us focus on the i-th feature of $\mathbf{z}$. We wish to find the first two moments, i.e., the mean and variance, of the marginal distribution $p\left(z_i|\mathcal{D}, \boldsymbol{\theta}_i, \beta_i\right)$. We can then push these moments through the pre-image map to obtain analytical solutions. This can be repeated for each i by virtue of the independence assumption. To begin, $p\left(z_i|\mathcal{D}, \boldsymbol{\theta}_i, \beta_i\right)$ is approximated as a Gaussian with mean m and variance v (Girard and Murray-Smith [2003]), which, from Appendix 4A, are given by:

$$m = \mathbb{E}_{\boldsymbol{\xi}}\left[\mathbf{c}_h(\boldsymbol{\xi}, \boldsymbol{\Xi}; \boldsymbol{\theta}_i)\right]^T \left(\mathbf{C}_i + \beta_i^{-1}\mathbf{I}\right)^{-1} \mathbf{z}_{:,i} \tag{4.23}$$

and:

$$\begin{aligned} v = {} & \mathbb{E}_{\boldsymbol{\xi}}\left[c_h(\boldsymbol{\xi}, \boldsymbol{\xi}; \boldsymbol{\theta}_i)\right] - m^2 \\ & - \left[\left(\mathbf{C}_i + \beta_i^{-1}\mathbf{I}\right)^{-1} - \left(\left(\mathbf{C}_i + \beta_i^{-1}\mathbf{I}\right)\mathbf{z}_{:,i}\right)^2\right] \mathbb{E}_{\boldsymbol{\xi}}\left[\mathbf{c}_h(\boldsymbol{\xi}, \boldsymbol{\Xi}; \boldsymbol{\theta}_i)^T \mathbf{c}_h(\boldsymbol{\xi}, \boldsymbol{\Xi}; \boldsymbol{\theta}_i)\right]. \end{aligned} \tag{4.24}$$

where $\mathbb{E}_{\boldsymbol{\xi}}[\cdot]$ and $\mathbb{V}\mathrm{ar}_{\boldsymbol{\xi}}(\cdot)$ are the expectation and variance with respect to $\boldsymbol{\xi}$, respectively. Calculation of these moments involves expectations of the kernel with respect

to the stochastic input distribution on the unknown and unobserved test inputs:

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\xi}}\left[c_h(\boldsymbol{\xi},\boldsymbol{\xi};\boldsymbol{\theta}_i)\right] &= \int c_h(\boldsymbol{\xi}',\boldsymbol{\xi}';\boldsymbol{\theta}_i)p(\boldsymbol{\xi}')d\boldsymbol{\xi}', \\
\mathbb{E}_{\boldsymbol{\xi}}\left[\mathbf{c}_h(\boldsymbol{\xi},\boldsymbol{\Xi};\boldsymbol{\theta}_i)\right] &= \int \mathbf{c}_h(\boldsymbol{\xi}',\boldsymbol{\Xi};\boldsymbol{\theta}_i)p(\boldsymbol{\xi}')d\boldsymbol{\xi}', \\
\mathbb{E}_{\boldsymbol{\xi}}\left[\mathbf{c}_h(\boldsymbol{\xi},\boldsymbol{\Xi};\boldsymbol{\theta}_i)^T\mathbf{c}_h(\boldsymbol{\xi},\boldsymbol{\Xi};\boldsymbol{\theta}_i)\right] &= \int \mathbf{c}_h(\boldsymbol{\xi}',\boldsymbol{\Xi};\boldsymbol{\theta}_i)^T\mathbf{c}_h(\boldsymbol{\xi}',\boldsymbol{\Xi};\boldsymbol{\theta}_i)p(\boldsymbol{\xi}')d\boldsymbol{\xi}'.
\end{aligned} \tag{4.25}$$

The analytic tractability of these integrals is dependent upon the choice of kernel and stochastic input distribution. One example of a kernel is the commonly used squared exponential, for which the integrals are derived in Appendix 4B. Once calculated, the mean can be pushed through the local pre-image mapping (4.15). Since we expect that the variance, on the other hand, will span more than one tangent space, predictions of the variance using this method may be inaccurate.

Since we cannot sample from the approximate marginal of the analytical approach, further analysis requires MC to fully characterize the distribution (4.22). Again it is sufficient to demonstrate the procedure for a single latent (feature space) dimension i. Using MC we obtain a marginalised predictive distribution expressed as the sum of normally distributed random variables, which itself is non-Gaussian:

$$\begin{aligned}
p(z_{\cdot,i}|\mathcal{D},\boldsymbol{\theta}_i,\beta_i) &= \int p(z_{\cdot,i}|\boldsymbol{\xi}',\mathcal{D},\boldsymbol{\theta}_i,\beta_i)p(\boldsymbol{\xi}')d\boldsymbol{\xi}' \\
&\simeq \frac{1}{Q}\sum_{q=1}^{Q} p(z_{\cdot,i}|\boldsymbol{\xi}^{(q)},\mathcal{D},\boldsymbol{\theta}_i,\beta_i) \\
&= \frac{1}{Q}\sum_{q=1}^{Q} \mathrm{N}(\mu(\boldsymbol{\xi}^{(q)}),\sigma^2(\boldsymbol{\xi}^{(q)})),
\end{aligned} \tag{4.26}$$

in which $\boldsymbol{\xi}^{(q)} \sim \mathrm{N}(\boldsymbol{\mu}_{\boldsymbol{\xi}},\Sigma_{\boldsymbol{\xi}})$, $\boldsymbol{\theta}_i$ and β_i are samples from the hyperparameter and signal noise (likelihood nugget) posteriors (for the i-th feature), and the approximation converges as $Q \to \infty$. Each sampled latent variable can then be pushed through the pre-image map. Latent variables found in this way are draws from the marginalised distribution $p\left(z_{\cdot,i}|\mathcal{D},\boldsymbol{\theta}_i,\beta_i\right)$ and we obtain multiple marginalised distributions (one for each $(\boldsymbol{\theta}_i,\beta_i)$) from which we can estimate the statistics of the response. Algorithm 8 describes the procedure. Note that we use a $*$ superscript in order to avoid confusion between MC samples and training points. Each $\mathbf{Y}_i^*$ in Algorithm 8 can be interrogated to find any property of the pushforward measure (mean, standard deviation and higher order moments). We can use kernel density estimation (KDE, also known as Parzen-Rosenblatt window, Simonoff [1996]) to ap-

proximate the probability density function given a finite number of samples, or find the moments of the density. We use Gaussian kernel function with a suitably small bandwidth.

Algorithm 8 Sample from the push forward measure

$S \leftarrow$ Number of hyperparameter posterior samples
$Q \leftarrow$ Number of draws from the input distribution $p(\boldsymbol{\xi})$
$\{\boldsymbol{\xi}_q^*\}_{q=1}^Q \leftarrow$ Dense set of Q draws from $p(\boldsymbol{\xi})$
for $s \leftarrow 1$ to S **do**
    $\boldsymbol{\Theta}_s$, $\boldsymbol{\beta}_s \leftarrow$ Sample from hyperparameter and signal precision posteriors
    for $q \leftarrow 1$ to Q **do**
        $\mathbf{z}_{s,q}^* \leftarrow$ Sample from (4.19) using $\boldsymbol{\Theta}_s$, $\boldsymbol{\beta}_s$, $\boldsymbol{\xi}_q^*$
        $\mathbf{y}_{s,q}^* \leftarrow$ Application of pre-image map (4.15) to $\mathbf{z}_{s,q}^*$
    end for
    $\mathbf{Y}_s^* \leftarrow [\mathbf{y}_{s,1}^* \dots \mathbf{y}_{s,Q}^*]^T$.
end for
Interrogate $\{\mathbf{Y}_s^*\}_{s=1}^S$ to extract statistics or distributions

4.4 Examples: Groundwater contamination

Groundwater contamination, caused by landfills, waste water seepage, hazardous chemical spillage, dumping of toxic substances or discharge from industrial processes (Karatzas [2017]), is a major concern for both public and environmental health. Understanding the mechanisms and predicting the transport of contaminants through soils is therefore an important topic in groundwater flow modelling.

The control of groundwater quality relies on knowledge of the transport of chemicals to the groundwater through soil. The efficacy of remedial treatment and management of contaminated land depends on the accuracy of models used for the simulation of flow and solute transport. Modelling and simulation of hydraulic phenomena in soil is, however, hampered by the complex and heterogeneous nature of soils, as well as the broad range of influential factors involved. A number of simplified models have been developed to describe the small-scale physical, chemical (Boi et al. [2009], Foo and Hameed [2009] and Vomvoris and Gelhar [1990]) and biological mechanisms (Schfer et al. [1998] and Barry et al. [2002]) that affect unsaturated flow and contaminant transport.

A current challenge in modelling solute transport in soils lies in characterising and quantifying the uncertainties engendered by the natural heterogeneity of the soil. Such uncertainty can be vital for decision making. Despite strong evidence from field-scale observations and experimental studies in relation to the ef-

fects of soil heterogeneity on the transport of contaminants (Al-Tabbaa et al. [2000] and Kristensen et al. [2010]), relatively few numerical models incorporate the effects of this uncertainty (Feyen et al. [1998], Aly and Peralta [1999], Sreekanth and Datta [2011b], Herckenrath et al. [2011] and Sreekanth and Datta [2014]).

Surrogate models have been used in a limited number of groundwater flow modelling problems (Aly and Peralta [1999], Bhattacharjya and Datta [2005], Kourakos and Mantoglou [2009], Sreekanth and Datta [2011a] and Ataie-Ashtiani et al. [2014]). We refer to Razavi et al. [2012] and Ketabchi and Ataie-Ashtiani [2015] for reviews of the topic. These are typically based on ANNs for approximating a small number of outputs within an optimization task. For example, Bhattacharjya and Datta used an ANN to approximate the salt concentration in pumped water at 8 pumping wells for 3 different times, in order to maximize the total withdrawal of water from a coastal aquifer while limiting the salt concentration (Bhattacharjya and Datta [2005]). Similarly, Kourakos and Mantoglou used an ANN model to optimize 34 well pumping rates in a coastal aquifer (Kourakos and Mantoglou [2009]). Monte Carlo has also been used in the context of groundwater flow modelling (Fu and Gomez-Hernandez [2009], Paleologos et al. [2006], Kourakos and Harter [2014], Maxwell et al. [2007] and Herckenrath et al. [2011]).

Our aim in this example is to develop a surrogate model for the values of a field variable in a groundwater flow model, e.g., the pressure, pressure head or flow velocity, at a high number of points in the spatial domain, in order to propagate uncertainty in a stochastic field input, e.g., the hydraulic conductivity. In such cases, simplified covariance structures (Conti and O'Hagan [2010]) for the output space (response surface) or dimensionality reduction for the input and/or output space can be used. Higdon et al. [2008] used principal component analysis (PCA) to perform linear, non-probabilistic dimensionality reduction on the response in order to render a GP model tractable (independent learning of a small number of PCA coefficients). Such linear approaches (PCA, multidimensional scaling, factor analysis) are applicable only when data lies in or near a linear subspace of the output space.

For more complex response surfaces, manifold learning (non-linear dimensionality reduction) can be employed, using for example kernel principal component analysis (kPCA), diffusion maps (Xing et al. [2016]) or Isomaps (Xing et al. [2015]). In contrast, kPCA was used to perform non-linear, non-probabilistic dimensionality reduction of the input space in Ma and Zabaras [2011]. This can be useful when the input space is generated from observations (experimental data), but when the form is specified we can use linear dimension reduction methods such as the

Karhunen-Loève expansion (KLE) (Wong [1971]).

4.4.1 Problem statement

Consider a well-defined, steady-state, partial differential equation (PDE) with a scalar, isotropic random field input (e.g., a permeability or hydraulic conductivity), and a response (output) consisting of a scalar field, e.g., pressure head, concentration or flow velocity. We may generalize our approach to multiple or vector fields but in order to simplify the presentation we focus on a single scalar field. We can also apply the method we develop to dynamic problems by focusing on the spatial field at a given fixed time (the second example we present). For an arbitrary input field realisation, solutions to the PDE are found using a numerical code (simulator, or solver) on a spatial mesh with k_y fixed degrees of freedom, e.g., grid points in a finite difference grid, control volume centres in a finite volume mesh, or spatial nodes in a finite element mesh combined with a nodal basis.

We denote the input field by $K(\mathbf{x})$, where $\mathbf{x} \in \mathcal{R} \subset \mathbb{R}^d$, $d \in \{1, 2, 3\}$ denotes a spatial location and the notation makes explicit the spatial dependence. The model output (a scalar field) is denoted by $u(\mathbf{x}; K)$, i.e., it is a function of $\mathbf{x}$ that is parametrised by $K(\mathbf{x})$. The random input $K(\mathbf{x})$ is defined on the whole of $\mathcal{R}$, and, therefore, requires a discrete (finite dimensional) approximation in order to obtain a numerical solution. Let $\mathbf{x}_k \in \mathcal{R}^d$, $k = 1, \ldots, k_y$ be a set of nodes or grid points and suppose that the simulator yields discrete approximations $\{u_k = u(\mathbf{x}_k; K)\}_{k=1}^{k_y}$ of the output field $u(\mathbf{x}; K)$ in each run. Our goal is to approximate these simulator outputs for an arbitrary K.

4.4.2 Input model: Karhunen-Loève expansion

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, with sample space Ω, event space $\mathcal{F}$ and probability measure $\mathbb{P}$. We can explicitly signify the randomness of the input by writing $K(\mathbf{x}, \omega)$, where $\omega \in \Omega$. For simplicity, and where it will not cause confusion, we suppress the dependence on ω (the same applies to other random processes). We assume that $K(\mathbf{x})$ is log-normal (to avoid unphysical, i.e., negative, realisations), so is of the form $K(\mathbf{x}, \omega) = \exp(Z(\mathbf{x}, \omega))$, where $Z(\mathbf{x}, \omega)$ is a normally distributed field (a GP[3] indexed by $\mathbf{x}$). For each $\mathbf{x} \in \mathcal{R}$, $Z(\mathbf{x}, \cdot) : \Omega \to \mathbb{R}$ is a random variable defined on the (common) probability space $(\Omega, \mathcal{F}, \mathbb{P})$. For a fixed $\omega \in \Omega$, $Z(\cdot, \omega) : \mathcal{R} \to \mathbb{R}$ is a deterministic function of $\mathbf{x}$ called a realization or sample path of the process. The

[3]Technically the process is a random field if the index (here $\mathbf{x}$) lies in $\mathbb{R}^L$ where $L > 1$ but the convention in the great majority of the literature is to use the term Gaussian *process* even in such cases.

mean and covariance functions of $Z(\mathbf{x}, \omega)$ are defined as:

$$\begin{aligned} m_Z(\mathbf{x}) &= \mathbb{E}[Z(\mathbf{x}, \omega)] = \int_\Omega Z(\mathbf{x}, \omega) d\mathbb{P}(\omega), \\ c_Z(\mathbf{x}, \mathbf{x}') &= \mathbb{E}\left[(Z(\mathbf{x}, \omega) - m_Z(\mathbf{x}))(Z(\mathbf{x}', \omega) - m_Z(\mathbf{x}'))\right], \end{aligned} \tag{4.27}$$

respectively, in which $\mathbb{E}[\cdot]$ is the expectation operator. Given the covariance and mean functions for $Z(\mathbf{x}, \omega)$, the most widely used finite dimensional approximation is based on a KLE (Wong [1971]). Assume that $Z(\mathbf{x}, \omega)$ is mean-square continuous ($\lim_{\delta\mathbf{x}\to\mathbf{0}} \mathbb{E}[(Z(\mathbf{x} + \delta\mathbf{x}, \omega) - Z(\mathbf{x}, \omega))^2] = 0$) and that $Z(\mathbf{x}, \omega) \in L^2(\mathcal{R} \times \Omega)$ ($\int_\mathcal{R} \mathbb{E}[|Z(\mathbf{x}, \omega)|^2] < \infty$), and is thus a second-order process. The Karhunen-Loève expansion theorem states that we may expresses $Z(\mathbf{x}, \omega)$ as a linear combination of deterministic $L^2(\mathcal{R})$-orthonormal functions $w_j(\mathbf{x})$, with random $L^2(\Omega)$-orthonormal coefficients $\xi_j(\omega)$:

$$Z(\mathbf{x}, \omega) = m_Z(\mathbf{x}) + \sum_{j=1}^{\infty} \sqrt{\lambda_j} \xi_j(\omega) w_j(\mathbf{x}), \tag{4.28}$$

where $\mathbf{x}$ are the spatial solver input variables; ξ are the coefficients; $\lambda_1 \geq \lambda_2 \geq \ldots > 0$ and $\{w_j(\mathbf{x})\}_{j=1}^{\infty}$ are respectively the eigenvalues and eigenfunctions of an integral operator with kernel $c_Z(\mathbf{x}, \mathbf{x}')$:

$$\int_\mathcal{R} c_Z(\mathbf{x}, \mathbf{x}') w_j(\mathbf{x}') d\mathbf{x}' = \lambda_j w_j(\mathbf{x}). \tag{4.29}$$

The random coefficients are given by:

$$\xi_j(\omega) = \frac{1}{\sqrt{\lambda_j}} \int_\mathcal{R} (Z(\mathbf{x}', \omega) - m_Z(\mathbf{x}')) w_j(\mathbf{x}') d\mathbf{x}', \tag{4.30}$$

and are independent, standard normal ($\xi_j \sim \mathrm{N}(0, 1)$), with $\mathbb{V}\mathrm{ar}(\sqrt{\lambda_j}\xi_j(\omega)) = \lambda_j$, where $\mathbb{V}\mathrm{ar}(\cdot)$ denotes the variance operator.

The sum (4.28) can be truncated by virtue of the decay in the eigenvalues for increasing j. Discretising the eigenvalue problem (4.29) using finite differencing at the nodes $\mathbf{x}_k \in \mathcal{R}$, $k = 1, \ldots, k_y$, assuming that they are uniformly distributed, leads to an eigenvalue problem for the covariance matrix $\mathbf{C} = [c_Z(\mathbf{x}_k, \mathbf{x}_j)]_{k,j=1}^{k_y}$:

$$\mathbf{C}_Z \mathbf{w}_j = \lambda_j \mathbf{w}_j, \tag{4.31}$$

where the k-th component $w_{j,k}$ of $\mathbf{w}_j \in \mathbb{R}^{k_y}$, $j = 1, \ldots, k_y$, is equivalent to the evaluation of eigenfunction w_j at the node $\mathbf{x}_k$, $k = 1, \ldots, k_y$. Defining the random

vector $\mathbf{Z} := (Z(\mathbf{x}_1), \ldots, Z(\mathbf{x}_{k_y}))^T : \Omega \to \mathbb{R}^{k_y}$, we can write:

$$\mathbf{Z} = \mathbf{m}_Z + \sum_{j=1}^{k_y} \sqrt{\lambda_j} \xi_j(\omega) \mathbf{w}_j, \tag{4.32}$$

where $\mathbf{m}_Z = (m_Z(\mathbf{x}_1), \ldots, m_Z(\mathbf{x}_{k_y}))^T$ and $\xi_j \sim \mathrm{N}(0,1)$ are independent random variables (note that we have kept the notation ξ_j and λ_j used in the continuous case in order to avoid notational clutter). This provides discrete realisations of $Z(\mathbf{x}, \omega)$ and the expansion in (4.32) can be truncated by virtue of the decay in λ_j for some $k_\xi < k_y$, chosen so that the generalised variance satisfies $\sum_{j=1}^{k_\xi} \sqrt{\lambda_j} / \sum_{j=1}^{k_y} \sqrt{\lambda_j} > \vartheta$ for some specified tolerance $0 < \vartheta < 1$. We can then obtain discrete realisations $\mathbf{K} = (K_1, \ldots, K_{k_y})^T$ of $K(\mathbf{x}, \omega)$ via:

$$K_k = K(\mathbf{x}_k, \omega) = \exp\left(m_Z(\mathbf{x}_k) + \sum_{j=1}^{k_\xi} \sqrt{\lambda_j} \xi_j(\omega) w_{j,k} \right). \tag{4.33}$$

The discrete input $\mathbf{K}$ can then replaced by the random vector defined by $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_{k_\xi})^T \sim \mathrm{N}(\mathbf{0}, \mathbf{I})$, the coefficients of which are independent standard normal. We may then write $u(\mathbf{x}_k; \boldsymbol{\xi})$ for the KLE approximation to $u(\mathbf{x}_k; K)$, at the nodes $\{\mathbf{x}_k\}_{k=1}^{k_y}$.

We note that different methods, including different quadrature rules or the use of projection schemes and Nystrom methods (Wan and Karniadakis [2006]) can be used to solve the eigenvalue problem (4.29), all of which lead to a generalised eigenvalue problem in place of (4.31) (Betz et al. [2014]). For example, if the finite element method is used, we may express the eigenfunctions as $w_j(\mathbf{x}) = \sum_k l_{j,k} \psi_k$ in terms of the finite element basis $\{\psi_k\}_{k=1}^{k_y}$ and perform a Galerkin projection of (4.29) onto $\mathrm{span}(\psi_1, \ldots, \psi_{k_y})$ to yield a generalised eigenvalue problem for $\{\lambda_j\}_{j=1}^{k_y}$ and the undetermined coefficients $\{l_{j,k}\}_{j,k=1}^{k_y}$ (Ghanem and Spanos [2003]).

The simulator (solver) can now be considered as a mapping $\boldsymbol{\eta} : \mathcal{X} \to \mathcal{Y}$ (assumed to be continuous and injective), where $\boldsymbol{\xi} \in \mathcal{X} \subset \mathbb{R}^{k_\xi}$ is the permissible *input space* and $\mathbf{y} \in \mathcal{Y} \subset \mathbb{R}^{k_y}$, is the permissible *output space* or *response surface* consisting of the discrete field:

$$\mathbf{y} = \boldsymbol{\eta}(\boldsymbol{\xi}) := (u(\mathbf{x}_1; \boldsymbol{\xi}), \ldots, u(\mathbf{x}_{k_y}; \boldsymbol{\xi}))^T. \tag{4.34}$$

4.4.3 Incorporating the Karhunen-Loève expansion

Our aim is to develop a surrogate to make fast, on-line predictions of $\boldsymbol{\eta}(\boldsymbol{\xi})$, using *training data* from a limited number of solver runs at the *design points* $\boldsymbol{\xi}_n,\ n = 1, \ldots, N$. The training data can be expressed compactly as a matrix $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]^T \in \mathbb{R}^{N \times k_y}$ and we can define $\boldsymbol{\Xi} = [\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_N]^T \in \mathbb{R}^{N \times k_\xi}$. The data set is thus $\mathcal{D}' = \{\boldsymbol{\Xi}, \mathbf{Y}\}$.

The high dimensionality of the input and output spaces pose great challenges for surrogate model development. The input space dimensionality can be reduced as described above. The intrinsic dimensionality of the output space is significantly lower than k_y by virtue of correlations between features for different inputs, as well as physical constraints imposed by the simulator. This suggests that we treat $\mathcal{Y}$ as a manifold and use manifold learning/dimensionality reduction to perform Bayesian inference on a low dimensional (feature) space $\mathcal{F}$ that is locally homeomorphic to $\mathcal{Y}$. Below we introduce the manifold learning method employed, before recasting the emulation problem as one of inference in the feature space, together with a pre-image (inverse) mapping to obtain solutions in $\mathcal{Y}$ for arbitrary inputs $\boldsymbol{\xi}$.

4.4.4 Predictive plots

We now assess the performance of the proposed method on two example partial differential equation problems: a Darcy flow problem with a contaminant mass balance, modelling steady state groundwater flow in a 2 dimensional porous medium; and Richards equation, modelling single phase flow through a 3 dimensional porous medium. As explained in section 4.3, the analysis includes: (i) predictions that are conditioned on an input; and (ii) predictions that are marginalised over the stochastic input.

When making conditioned predictions, we use the conditional predictive distribution (4.20) for $\mathbf{y}$, or the distribution (4.19) for $\mathbf{z}$ in conjunction with the pre-image map (4.15). As explained in section 4.2, we place a prior over the hyper-parameters $\boldsymbol{\Theta}$ and signal variances $\boldsymbol{\beta}$ and use a HMC scheme to sample from the posterior distributions over these parameters. Each sample can be used to obtain a different normal predictive distribution, conditioned on an input. We are, therefore, able to see how the predictive mean and variance change with respect to the uncertainty in the GP parameters. In the results, we plot the expectation and standard deviation of the first two predictive distribution moments.

For the forward UQ problem, we marginalize the conditional predictive distributions over a stochastic input (4.22) to obtain the pushforward measure (non

analytically). We are able to analytically find the mean using (B.2) and (B.3) together with the pre-image map, or, using Algorithm 8, sample from the marginalised distribution *via* MC (4.26).

The accuracies of both the point predictions and the predictions of the push-forward measure are assessed by comparison with the true values obtained with the simulator (on the test inputs $\{\boldsymbol{\xi}_q^*\}_{q=1}^Q$). We run the solver for each test input to generate the true response, denoted $\tilde{\mathbf{y}}_q^*$. For the UQ comparison we again approximate the pdf using KDE (or simply extract the moments) based on $\{\tilde{\mathbf{y}}_q^*\}_{q=1}^Q$. The latter approximation is guaranteed to converge to the truth as the number of test inputs increases.

4.4.5 Case 1: Darcy Flow, non-point source pollution

The first example is a linear model of steady state groundwater flow in 2 dimensions. The approach was developed in Kourakos et al. [2012] and implemented in the mSim package[4]. The model comprises Darcy's law and a contaminant mass balance in a 2-d polygonal domain Ω of total area 18.652 km^2 containing wells and a stream, and subdivided into polygonal regions of different land use (Fig. 4.6). Full details of the model and the numerical method can be found in Kourakos et al. [2012]. Below we provide a brief description. The model equations are given by:

$$\begin{aligned} \nabla \cdot (K \nabla h) &= Q \\ R\frac{\partial C}{\partial t} &= \nabla \cdot (\mathbf{D} \nabla C) - \nabla \cdot (\mathbf{v} C) = G \end{aligned} \tag{4.35}$$

in which $K(\mathbf{x})$ is the hydraulic conductivity, $h(\mathbf{x})$ is the pressure head, $C(\mathbf{x}, t)$ is the contaminant concentration, R is the retardation factor, $\mathbf{D}$ is the dispersion tensor, $\mathbf{v}$ is the fluid velocity, and Q and G represent sources/sinks. The contaminant transport equation is replaced by a 1-d approximation and is solved through an ensemble of one dimensional streamline-based solutions (Kourakos et al. [2012]).

The contaminant balance and flow (Darcy) equations are decoupled. The latter is solved using the finite element method based on triangular elements and first order (linear) shape functions. The boundary conditions are given by: (i) a constant head equal to 30 m on the left boundary; (ii) a general head boundary equal to 40 m with conductance equal 160 m^3 day^{-1} on the right boundary; and (iii) no flow on the top and bottom boundaries. Each land use polygon is assigned

[4]See `http://subsurface.gr/joomla/msim_doc/twoD_examples_help.html` for full details of the implementation, including the domain, mesh generation and boundary conditions. Last accessed 29 August 2017.

its own recharge rate. Stream rates are assigned directly to nodes (any node closer than 10 m to the stream is considered to be part of the stream).

We assume that $K(\mathbf{x})$ is log-normally distributed and treat it as an input. The output field upon which we focus is the pressure head, that is, $u(\mathbf{x}; K) = h(\mathbf{x})$ in the notation of section 4.4.1. We use the input model described in section 4.4.2, defining a discretised random field corresponding to realisations of $K(\mathbf{x}) = \exp(Z(\mathbf{x}))$ at the nodes $\{\mathbf{x}_k\}_{k=1}^{k_y} \subset \mathcal{R}$ on the finite element mesh. The covariance function for the random field $Z(\mathbf{x})$ is given by:

$$c_Z(\mathbf{x}, \mathbf{x}') = \sigma_Z^2 \exp\left\{-\frac{(x_1 - x_1')^2}{l_1^2} - \frac{(x_2 - x_2')^2}{l_2^2}\right\}, \quad \mathbf{x} = (x_1, x_2)^T \in \mathcal{R}, \tag{4.36}$$

in which l_1 and l_2 are correlation lengths. This separable form was suggested in Zhang and Lu [2004] and is used extensively in the literature to model hydraulic permeability fields (often by setting the correlation lengths equal). The generalised variance (value of k_ξ) was chosen to satisfy $\sum_{j=1}^{k_\xi} \sqrt{\lambda_j} / \sum_{j=1}^{k_y} \sqrt{\lambda_j} > 0.98$.

Both the training and test input samples were drawn independently: $\boldsymbol{\xi}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $n = 1, \ldots, N$ to yield $\{\mathbf{y}_n\}_{n=1}^N$ for training; and $\boldsymbol{\xi}_q \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $q = 1, \ldots, Q$ to yield $\{\widetilde{\mathbf{y}}_q^*\}_{q=1}^Q$ for testing. We set $Q = 5000$ and $N \in \{25, 50, 75, 100\}$. Running the solver with an input generated using the KLE truncation necessarily leads to a response surface with intrinsic dimension at most k_ξ, which was, therefore, the value chosen for the approximating manifold dimension k_z. In all of the results presented below, $k_y = 1933$ nodes (elements) were used in the simulation. The number of neighbours P in the LTSA algorithm was chosen according to the error between the solver response and the predictive mean at the test points. We define a scaled measure of error on each test point as follows:

$$e_q = ||\widetilde{\mathbf{y}}_q^* - \overline{\mathbf{y}}_q^*|| / ||\widetilde{\mathbf{y}}_q^*||, \quad q = 1, \ldots, Q, \tag{4.37}$$

in which $\widetilde{\mathbf{y}}_q^*$ is the response predicted by the solver, and $\overline{\mathbf{y}}_q^*$ is the point recovered by application of the preimage map (4.15) on the GP *predictive mean* (4.18). The scaling ensures that the errors are comparable and can be interpreted as percentage errors.

We present results for three stochastic input models:

M1 We set $m_Z = \ln(40)$ and $\sigma_Z^2 = 0.2$, yielding[5] a mean for $K(\mathbf{x})$ of 44.2 m day^{-1}, which is close to the default value in the mSim package, and a standard

[5] If $Z(\mathbf{x})$ has a mean and variance of μ and ν, then the mean and variance of the lognormal process $\exp(Z(\mathbf{x}))$ are $\mu' = \exp(\mu + \nu/2)$ and $\nu' = \exp(2\mu + \nu)(\exp(\nu) - 1)$, respectively.

deviation of 13.63 m day^{-1}. The correlation lengths were chosen as $l_1 = 2000$ m and $l_2 = 1000$ m, which correspond to dimensionless values of 1/3 and 2/7 respectively. These choices require $k_\xi = 5$ input dimensions to capture 98% of the generalised variance.

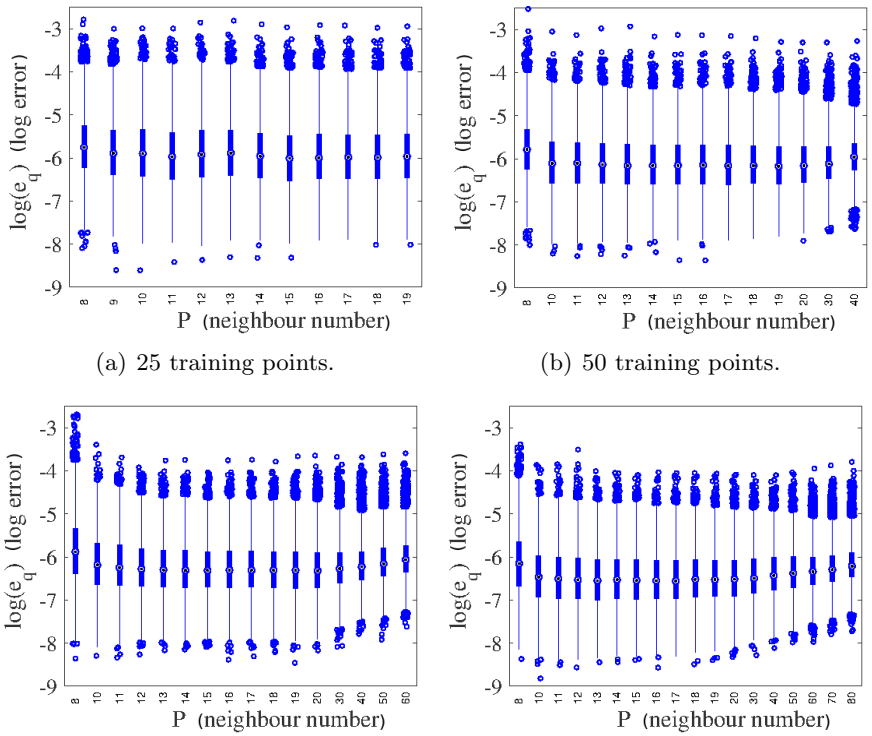
M2 We set $m_Z = \ln(36.18)$ and $\sigma_Z^2 = 0.4$, again yielding a mean 44.2 m day^{-1} and a standard deviation of 18.80 m day^{-1}. We set $l_1 = 2000$ m and $l_2 = 1000$ m. $k_\xi = 5$ captures 98% of the generalised variance.

M3 We set $m_Z = \ln(40)$, and $\sigma_Z^2 = 0.4$ and reduce the correlation lengths to $l_1 = 1000$ m and $l_2 = 500$ m (dimensionless values of 1/6 and 1/7 respectively). We now require $k_\xi = 15$ to capture 98% of the generalised variance.

For model **M1**, the distributions of $\{e_q\}_{n=1}^Q$ for different training set sizes N are shown as boxplots for increasing values of P in Fig. 4.2. The performance of the emulator is good even for $N = 25$ training points (maximum e_q of approximately e^{-3}), although there is a clear decrease in the error when N is increased to 100. The relationship between the errors and P is more complicated. The errors are high for $P < 8$ (not shown in the boxplots) at all values of N and decrease as P increases. This is due to the linear approximation of points in local tangent spaces *via* PCA in the LTSA algorithm. As more points are added, the approximation improves. As P is increased beyond a certain value, however, the errors increase (this is most clearly visible for $N = 100$). The reason for this behaviour is that, for large enough neighbourhood sizes, the linear approximation breaks down. Thus, there is an optimal choice of P for each value of N and the higher the value of N the more sensitive are the errors to the value of P. In the subsequent results we use $P = 15$ unless otherwise specified.

In Fig. 4.3 we plot the *normalised* pressure head prediction (for each co-ordinate of the predicted pressure head we subtract the mean and divide by the standard deviation) corresponding to the highest e_q for both $N = 25$ and $N = 50$ (using $P = 15$). The normalisation highlights the differences between the true values and the predictions (the errors) more clearly. The predicted means of the means (middle row) are the mean predictions averaged over all hyperparameter and precision samples. Also shown (bottom row) are the standard deviations of the predictions averaged over all hyperparameter and precision samples. We observe that the prediction at $N = 75$ is highly accurate, while the prediction at $N = 25$ is still reasonably accurate even in this worst case (an outlier in Fig. 4.2). For both values of N, the true values lie within the credible regions. In Fig. 4.4 we show the corresponding predictions for cases where the errors are close to the medians.

Both predictions are highly accurate and again the true values lie inside the credible regions.



(a) 25 training points.

(b) 50 training points.

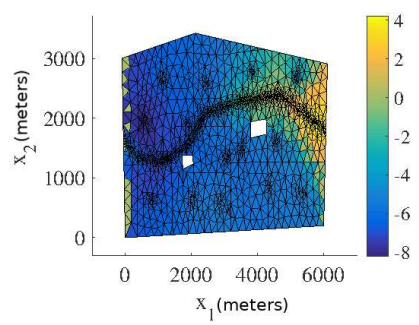
(c) 75 training points.

(d) 100 training points.

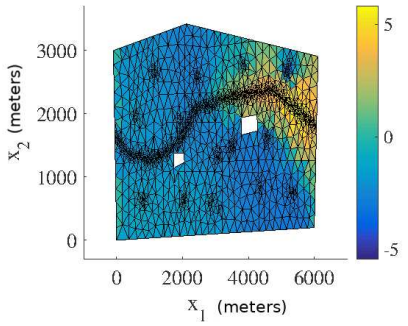
Figure 4.2: Log normalised error $\ln(e_q)$ in the normalised pressure head prediction for an emulator trained on $N = 25$, 50, 75 and 100 points $\mathbf{y}_n$ and tested with $Q = 5000$ test points $\widetilde{\mathbf{y}}_q^*$ for different nearest neighbour numbers P (model **M1**). Predictions were obtained by averaging over hyperparameter and precision posterior samples.

We now focus on the forward problem, in which we estimate the marginalised predictive distribution (4.22) using Algorithm 8. KDE is used to obtain estimates of the pdf of an output feature for different predictive posterior, hyperparameter and precision samples, as previously described. The output feature we choose is the pressure head at the spatial location $\mathbf{x} = (2511, 486) \in \mathcal{R}$. We plot a heat map of the pdfs in Fig. 4.5 for different N.

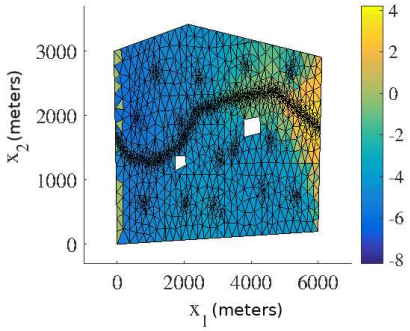
The distributions are accurately estimated for all values of N. While the predictions improve as the number of training samples N increases, the true value does not always lie within the contours. This is because: (i) as stated earlier, an increased GP predictive variance acts to smooth the density, rather than increase the width of the contours; (ii) by choosing *a priori* the number of neighbours we also



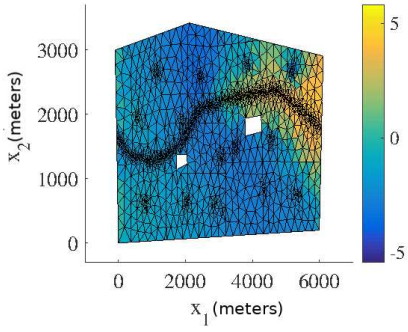
(a) True value, $N = 25$.



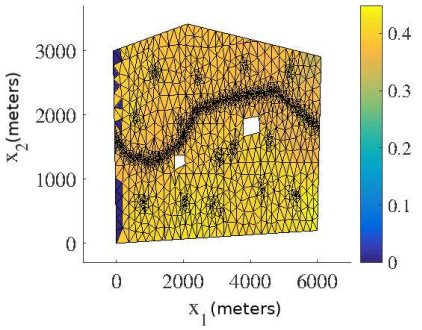
(b) True value, $N = 75$.



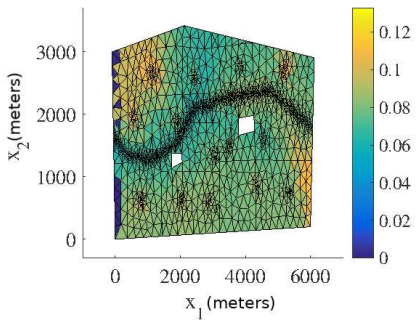
(c) Mean of the mean, $N = 25$.



(d) Mean of the mean, $N = 75$.



(e) Mean of the standard deviation, $N = 25$.

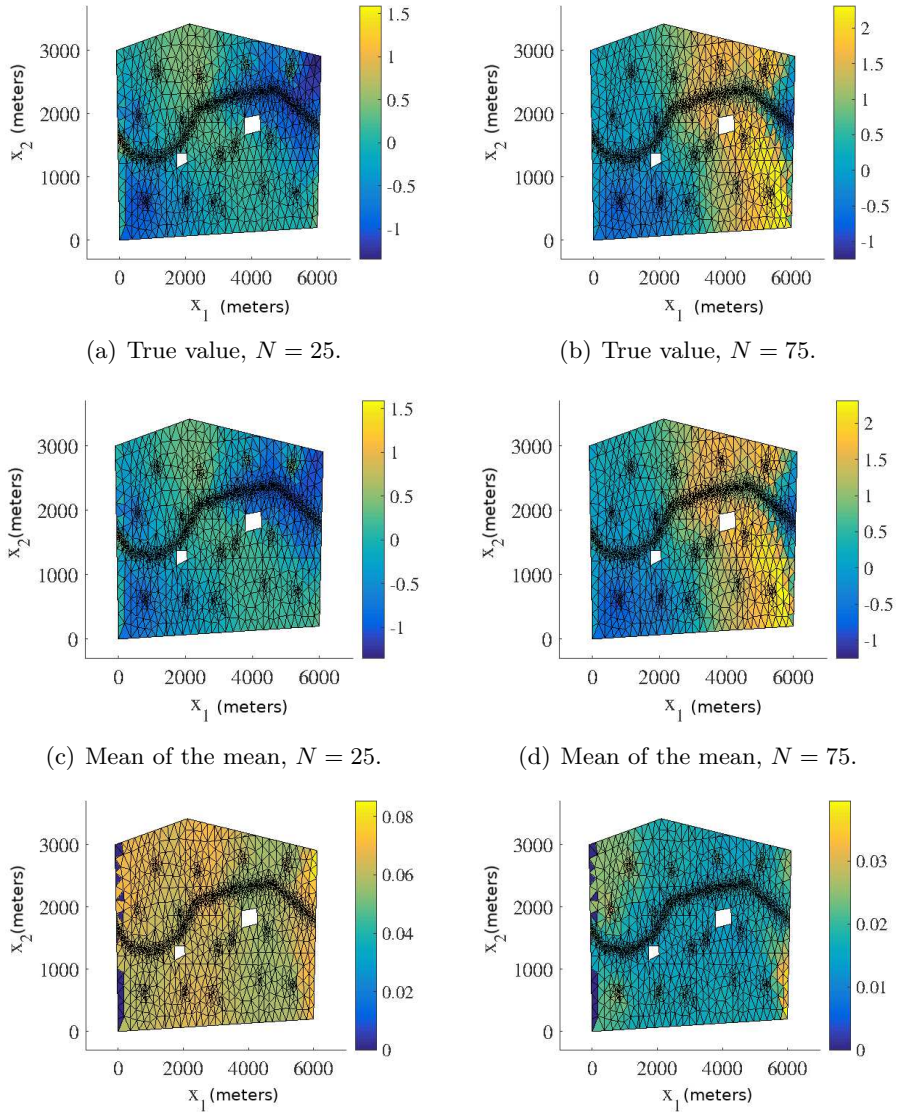


(f) Mean of the standard deviation, $N = 75$.

Figure 4.3: The test predictive mean and standard deviation of the normalised pressure head for the point with highest error from an emulator using $P = 15$, corresponding to the relevant boxplot in Fig. 4.2, for both 25 and 75 training points (model **M1**).

a priori assume a *global* smoothness of the emulator; and (iii) we have a pre-image map $\widehat{\mathbf{f}} : \mathcal{F} \to \mathcal{Y}$ for which the error is unknown (as with all methods), but not estimated (as with probabilistic methods).

We can find the means and standard deviations across the samples obtained



(a) True value, $N = 25$.

(b) True value, $N = 75$.

(c) Mean of the mean, $N = 25$.

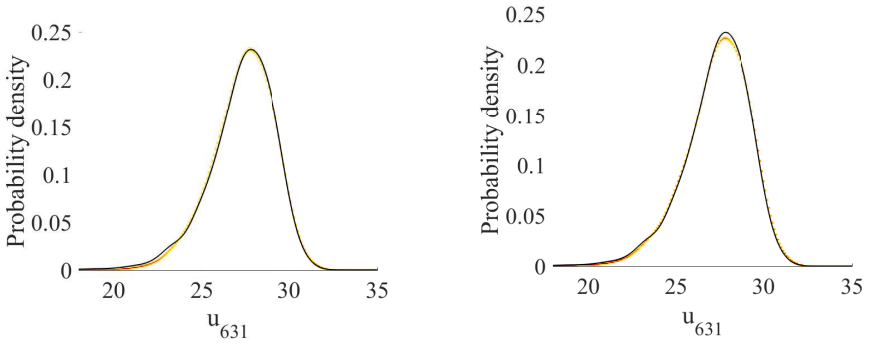
(d) Mean of the mean, $N = 75$.

(e) Mean of the standard deviation, $N = 25$.

(f) Mean of the standard deviation, $N = 75$.

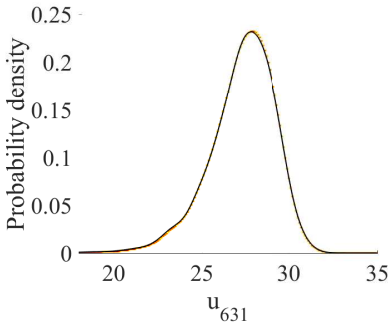
Figure 4.4: The test predictive mean and standard deviation of the normalised pressure head for a test point with an error near the median of the $P = 15$ boxplot in Fig. 4.2 from emulators using $P = 15$, for both 25 and 75 training points (model **M1**).

for different predictive posterior, hyperparameter and precision samples using Algorithm 8. We obtain distributions over the moments of the marginalised predictive distribution (4.22). In Fig. 4.6 we plot the mean and standard deviation of the marginalised predictive mean and standard deviation for $N = 25$, with comparisons

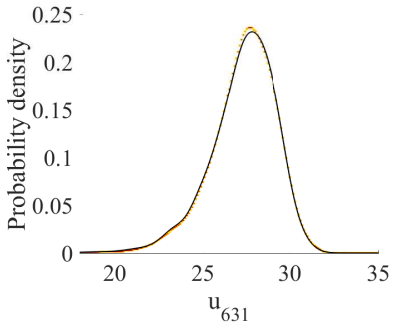


(a) 25 training points, 15 k-NN.

(b) 50 training points, 15 k-NN.



(c) 75 training points, 15 k-NN.



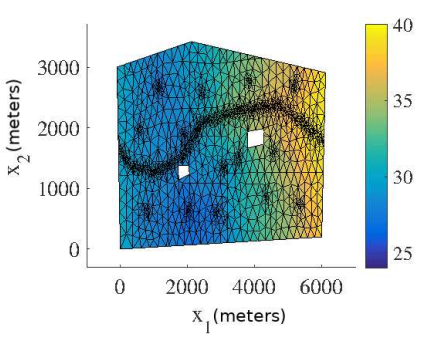
(d) 100 training points, 15 k-NN.

Figure 4.5: The pdfs of the pressure head response at the spatial coordinate $\mathbf{x} = \mathbf{x}_{631} = (2511, 486) \in \mathcal{R}$ on the finite-difference grid, obtained using kernel density estimation on $Q = 5000$ points (Model **M1**). The black line gives the MC prediction using the simulator. The contours show how the emulator predictions vary with hyperparameter, precision and predictive distribution samples.

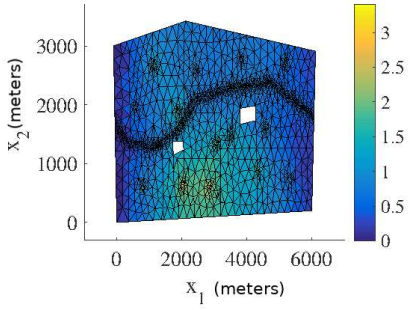
to the true values obtained by finding the mean and standard deviation across the test responses $\{\widetilde{\mathbf{y}}_q^*\}_{q=1}^Q$. Even for this low number of training points the results are highly accurate.

We now consider Model **M2**, in which we increase the variance of the stochastic input, while keeping the mean fixed. For this example we again set $l_1 = 2000$ m and $l_2 = 1000$ m, requiring $k_\xi = 5$. The distributions of $\{e_q\}_{q=1}^Q$ for different training set sizes N and increasing P are shown in Fig. 4.7. We observe trends similar to those observed using Model **M1**, although the increased variance leads to larger errors at fixed N and P (higher maxima and minima). With the exception of an isolated outlier (shown later), the predictions are nevertheless accurate for $N = 75$.

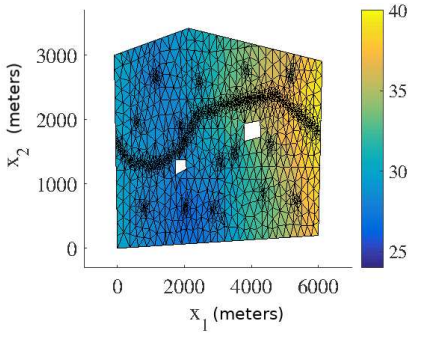
The worst case (highest e_q) for $P = 15$ is shown in Fig. 4.8 for $N = 25$ and 75 points (see Fig. 4.7). As before, the top row is the test (solver prediction), while



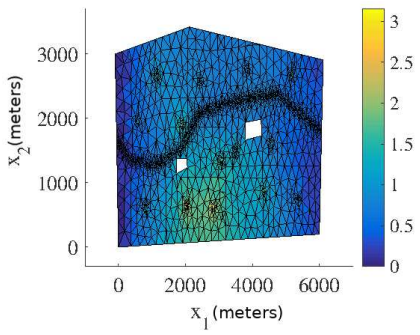
(a) Monte Carlo mean.



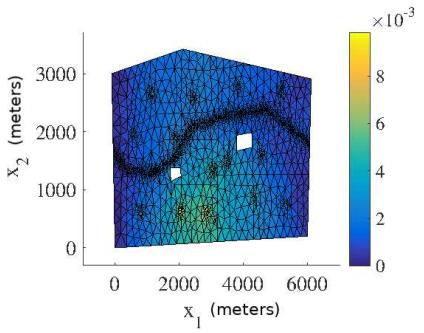
(b) Monte Carlo standard deviation.



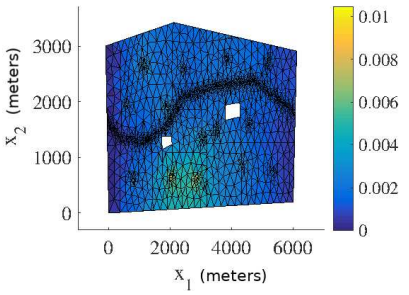
(c) Mean of the mean for 25 training points.



(d) Mean of the standard deviation for 25 training points.



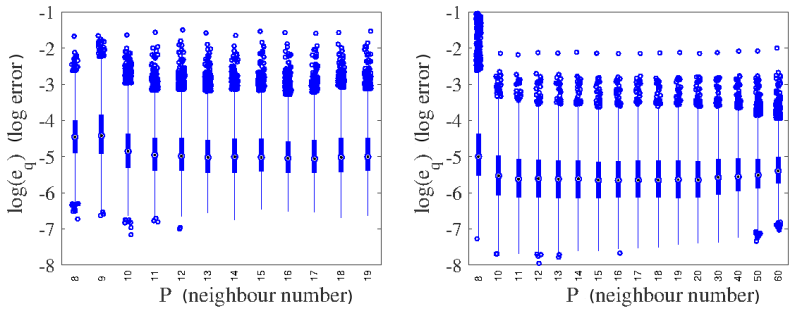
(e) Standard deviation of the mean for 25 training points.



(f) Standard deviation of the standard deviation for 25 training points.

Figure 4.6: Moments of the mean and standard deviation of the pressure head in Model **M1**. The emulator variation is a consequence of the hyperparameter, precision and predictive distribution samples. We have a single, parametrised realisation of the manifold.

the middle and bottom rows are the mean prediction and standard deviation of the prediction averaged over all hyperparameter and precision samples. The true values lie within the credible regions, although for this model a higher number of training



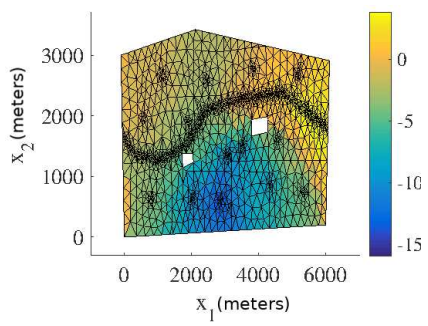
(a) 25 training points. (b) 75 training points.

Figure 4.7: Log normalised error $\ln(e_q)$ in the normalised pressure head prediction for an emulator trained on $N = 25$ and 75 points $\mathbf{y}_n$ and tested with $Q = 5000$ test points $\widetilde{\mathbf{y}}_q^*$ for different nearest neighbour numbers P (model **M2**). Predictions were obtained by averaging over hyperparameter and precision posterior samples.

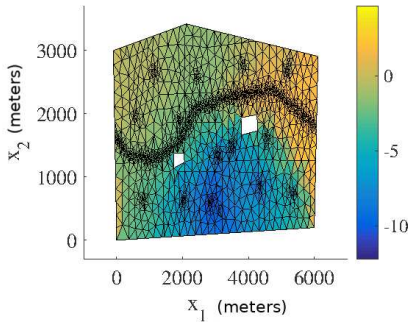
points are required to ensure that even the worst case predictions are accurate. Fig. 4.9 demonstrates the quality of the predicted responses when the errors are at the median in the $P = 15$ boxplots in Fig. 4.7. Here, even $N = 25$ provides accurate results.

Fig. 4.10 shows heat maps of the pdfs of the pressure head at the spatial location $\mathbf{x} = (2511, 486)$ for different N (generated using KDE) in the case of Model **M2**. Using $N = 75$ we achieve very good agreement with the MC prediction based on the simulator results (test points), although again the true value does not lie within the contours. For $N = 25$, we plot the mean and standard deviation of the marginalised predictive mean and standard deviation in Fig. 4.11, with a comparison to the true values obtained from $\{\widetilde{\mathbf{y}}_q^*\}_{q=1}^Q$. The predictions are highly accurate. In fact, even for $N = 25$ (not shown to conserve space) the mean was very accurate and the standard deviation exhibited only slight differences from the true value.

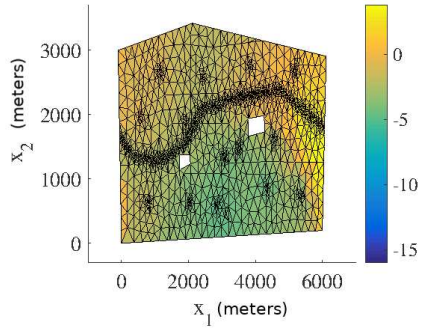
For Model **M3** (decreased correlation lengths, high standard deviation and $k_\xi = 15$), the distributions of $\{e_q\}_{q=1}^Q$ for increasing N and P in are shown in Fig. 4.12. In this case it is clear that a much higher value of P ($P > 60$, giving a similar neighbourhood radius in-line with the increased sample density) is required to obtain a reasonable accuracy. For $N = 500$ and $P = 80$, there are a small (9 out of 5000) number of outliers with low accuracy, while the errors for the remaining points satisfy $\ln(e_q) < -3.25$. The worst cases (highest e_q) for $P = 70$, $N = 300$ and $P = 80$, $N = 500$ are shown in Fig. 4.13, and in Fig. 4.14 we show predicted responses with errors at the medians for the same values of P and N. There are noticeable differences in the worst cases, although the qualitative agreement is very



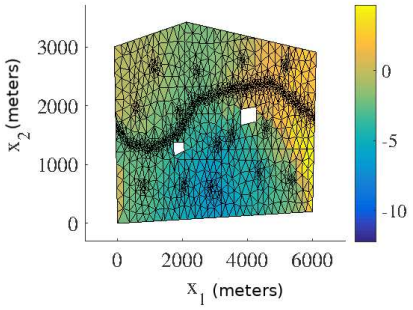
(a) True value, $N = 25$.



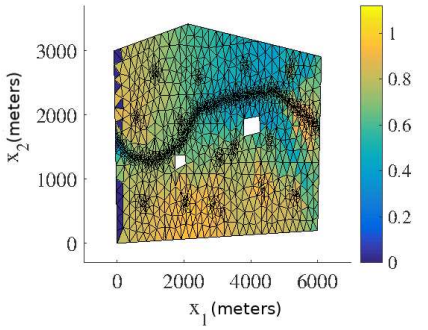
(b) True value, $N = 75$.



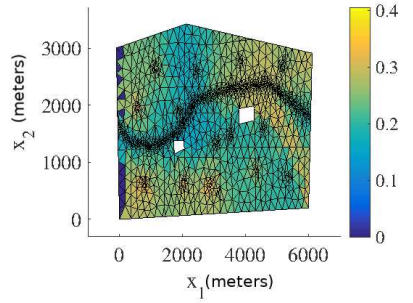
(c) Mean of the mean, $N = 25$.



(d) Mean of the mean, $N = 75$.



(e) Mean of the standard deviation, $N = 25$.

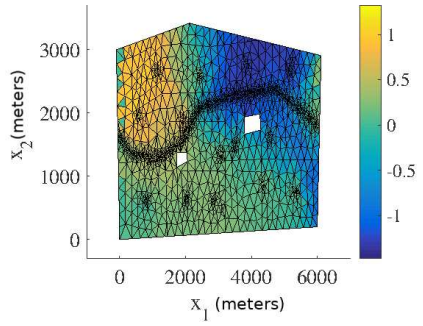


(f) Mean of the standard deviation, $N = 75$.

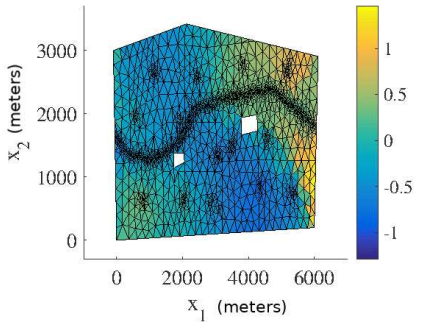
Figure 4.8: The test predictive mean and standard deviation of the normalised pressure head in the case of the highest errors e_q in Fig. 4.7 for $P = 15$ and $N = 25$ and 75 training points (Model **M2**).

good at both values of N. For the median error cases both emulators perform extremely well.

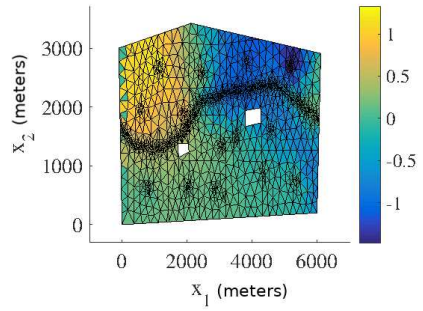
In Fig. 4.15 we show the heat maps of the pdfs of the pressure head at $\mathbf{x} = (2511, 486)$ for different N. For both values of N there is very good agreement with the simulator result and the true value this time lies within the contours. For



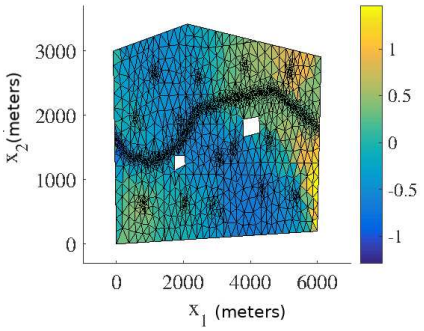
(a) True value, $N = 25$.



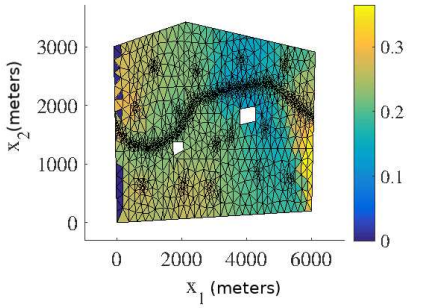
(b) True value, $N = 75$.



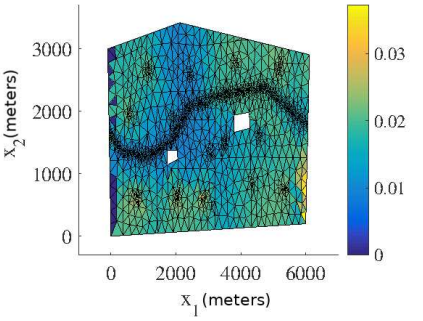
(c) Mean of the mean, $N = 25$.



(d) Mean of the mean, $N = 75$.



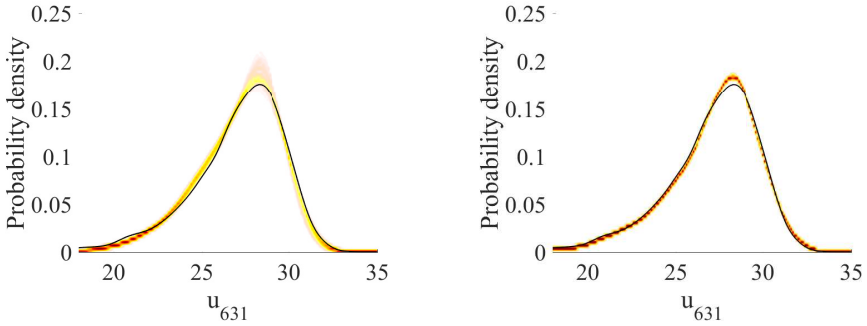
(e) Mean of the standard deviation, $N = 25$.



(f) Mean of the standard deviation, $N = 75$.

Figure 4.9: The test predictive mean and standard deviation of the normalised pressure head in the case of errors e_q near the median in Fig. 4.7 for $P = 15$ and $N = 25$ and 75 training points (Model **M2**).

$N = 500$, we show the mean and standard deviation of the marginalised predictive mean and standard deviation in Fig. 4.16, with a comparison to the true values obtained from $\{\widetilde{\mathbf{y}}_q^*\}_{q=1}^Q$. The predictions are again highly accurate (which was also the case for $N = 300$).



(a) 25 training points, 15 k-NN. (b) 75 training points, 15 k-NN.

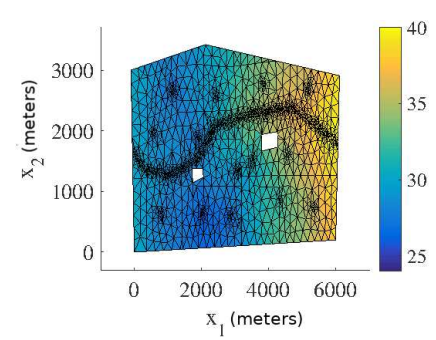
Figure 4.10: The pdfs of the pressure head response at the spatial coordinate $\mathbf{x} = \mathbf{x}_{631} = (2511, 486)$ on the finite-difference grid, obtained using kernel density estimation on $Q = 5000$ points (Model **M2**). The black line gives the MC prediction using the simulator. The contours show how the emulator predictions vary with hyperparameter, precision and predictive distribution samples.

4.4.6 Case 2: Richards equation, unsaturated flow in porous media

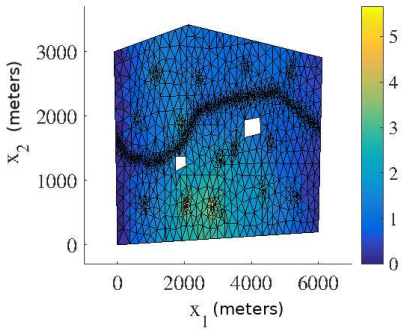
Consider a single-phase flow through a 3 dimensional porous region $\mathcal{R} \subset \mathbb{R}^3$ containing unsaturated soil with a random permeability field. The vertical flow problem can be solved using Richards equation (Darcy's law combined with a mass balance). There are three standard forms of Richards equation: the pressure head based (h-based) form; the water content-based (θ-based) form; and the mixed-based form. For flow in saturated or layered soils, the h-based form is particularly appropriate (Huang et al. [1996] and Shahraiyni and Ataie-Ashtiani [2011]).

The h-based form with an implicit or explicit finite difference (FD) scheme has been shown to provide good accuracy, although this approach may result in high mass balance errors (Zarba et al. [1990] and Huang et al. [1996]). The mixed-based form, on the other hand, exhibits low mass balance errors with highly accurate predictions using a fully implicit FD scheme (Ray and Mohanty [1992], Zarba et al. [1990] and Celia et al. [1987]). The latest work of Shahraiyni and Ataie-Ashtiani [2011] showed that a fully implicit FD scheme with a standard chord slope (CSC) approximation (Rathfelder and Abriola [1994]) not only solved the mass balance problem of the h-based form but also improved convergence. Thus, in the paper we adopt this approach, although other numerical formulations are by no means precluded. The h-based form of Richards equation can be written as follows:

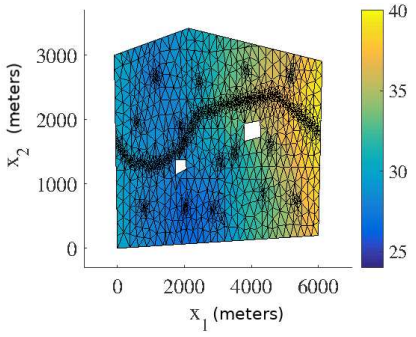
$$u(h)\frac{\partial h}{\partial t} - \nabla \cdot K(h)\nabla(h + x_3) = 0, \qquad (\mathbf{x}, t) \in \mathcal{R} \times (0, T], \tag{4.38}$$



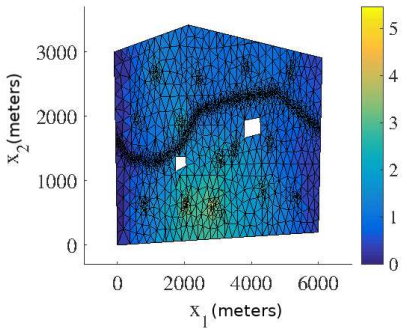
(a) Monte Carlo mean.



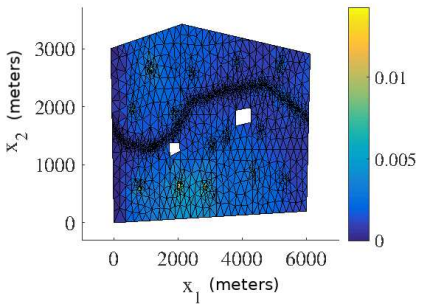
(b) Monte Carlo standard deviation.



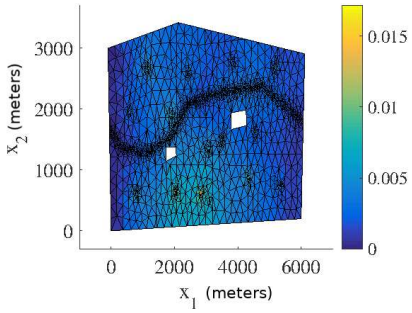
(c) Mean of the mean for 75 training points.



(d) Mean of the standard deviation for 75 training points.



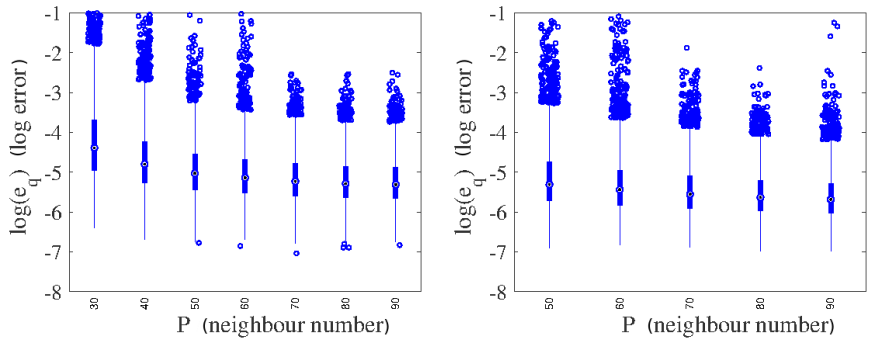
(e) Standard deviation of the mean for 75 training points.



(f) Standard deviation of the standard deviation for 75 training points.

Figure 4.11: Moments of the mean and standard deviation of the pressure head for $P = 15$, $N = 75$ (Model **M2**). The emulator variation is a consequence of the hyper-parameter and predictive distribution samples. We have a single, parametrised realisation of the manifold.

where h is the pressure head, $u(h) = \partial\theta/\partial h$ is the specific moisture capacity, in which θ is the moisture content, $K(h)$ is the unsaturated hydraulic conductivity, and $\boldsymbol{x} = (x_1, x_2, x_3)^T$ is the spatial coordinate, in which x_3 is the vertical coordinate.



(a) 300 training points. (b) 500 training points.

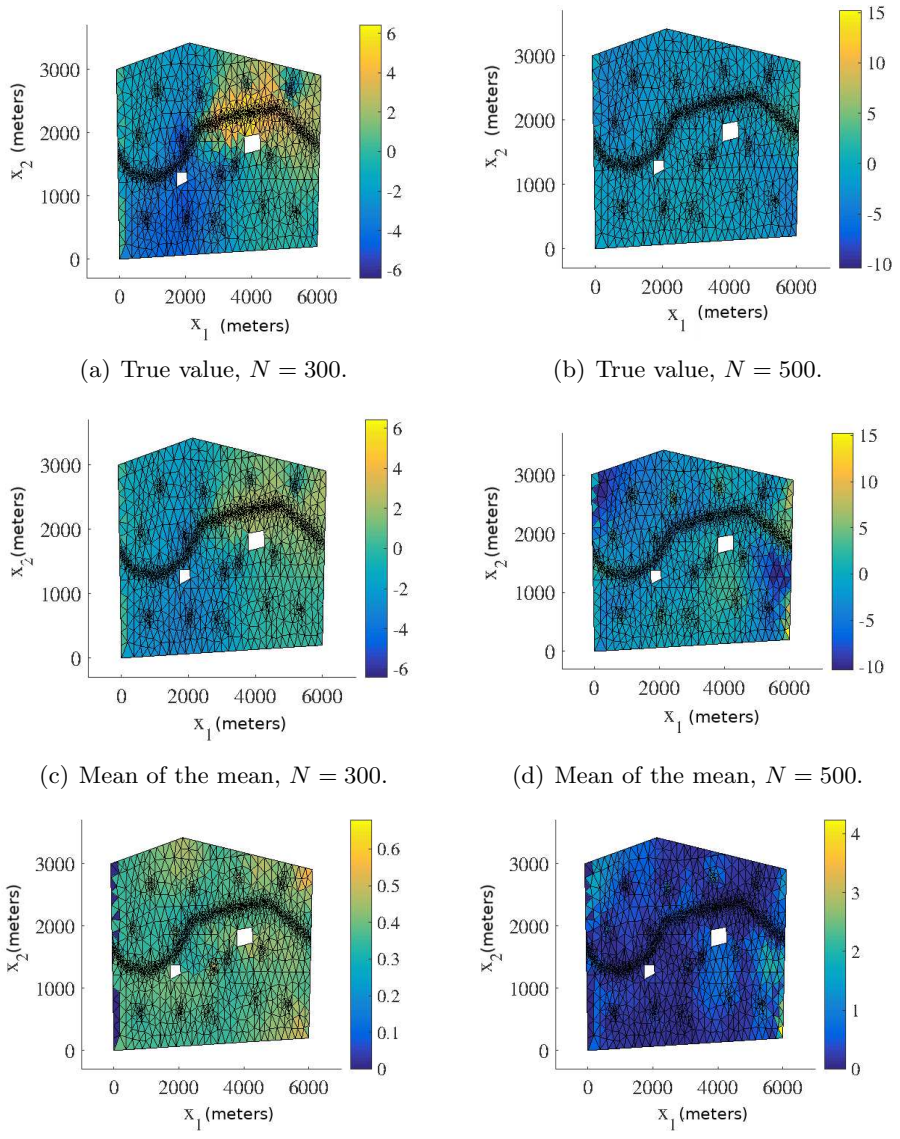
Figure 4.12: Log normalised error between true and predictive mean in the normalised pressure head prediction from an emulator trained on 300 and 500 points $\mathbf{y}_n$ and interrogated with $Q = 5000$ test points $\tilde{\mathbf{y}}_q^*$ for different nearest neighbour numbers P (Model **M3**). Predictions were obtained by averaging over hyperparameter and precision posterior samples.

The non-linear functions $\theta(h)$ and $k(h)$ can take on different forms. For example, in Haverkamp *et al.* Haverkamp et al. [1977], a least square fit to experimental data was used to derive:

$$\begin{aligned} \theta(h) &= \frac{\alpha_1(\theta_s - \theta_r)}{\alpha_1 + |h|^{\alpha_2}} + \theta_r, \\ K(h) &= K_s(\mathbf{x})\frac{\alpha_3}{\alpha_3 + |h|^{\alpha_4}}, \end{aligned} \tag{4.39}$$

where θ_r and θ_s are the residual the saturated water contents, $K_s(\mathbf{x})$ is the saturated hydraulic conductivity, and α_1, α_2, α_3 and α_4 are fitting parameters. We adopt the relationships (4.39) and use the parameter values in Haverkamp et al. [1977]: $\alpha_1 = 1.611 \times 10^6$, $\alpha_2 = 3.96$, $\alpha_3 = 1.175 \times 10^6$, $\alpha_4 = 4.74$, $\theta_s = 0.287$ and $\theta_r = 0.075$. The domain $\mathcal{R}$ is taken to be 20 cm $\times$ 20 cm $\times$ 20 cm. $K_s(\mathbf{x})$ is treated as a random field input with a log-normal distribution ($K_s(\mathbf{x}) = \exp(Z(\mathbf{x}))$, again discretised using the Karhunen-Loève theorem. We generate realisations of a corresponding discrete random field on an $n_1 \times n_2 \times n_3$ finite difference grid ($k_y = n_1 n_2 n_3$), with grid spacings Δx_1, Δx_2 and Δx_3 in the directions x_1, x_2 and x_3, respectively. The output field of interest is again the pressure head, at a fixed time T. Thus, we set $u(\mathbf{x}; K) = h(\mathbf{x}, T)$.

The boundary conditions are those used in Haverkamp et al. [1977], corresponding to laboratory experiments of infiltration in a plexiglass column packed with sand. Along the top boundary (surface) $x_3 = 20$ cm, the pressure head is



(a) True value, $N = 300$.

(b) True value, $N = 500$.

(c) Mean of the mean, $N = 300$.

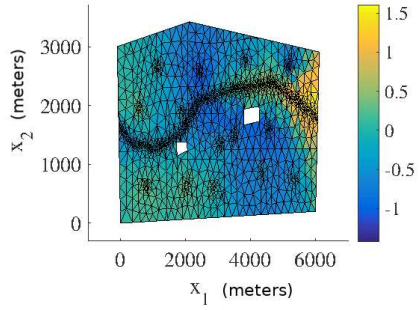
(d) Mean of the mean, $N = 500$.

(e) Mean of the standard deviation, $N = 300$.

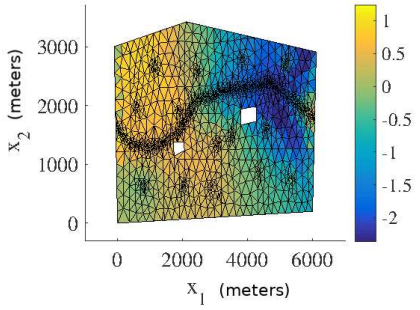
(f) Mean of the standard deviation, $N = 500$.

Figure 4.13: The test predictive means and standard deviations for predictions of the normalised pressure head with the highest errors from emulators using $P = 70$, $N = 300$ and with $P = 80$, $N = 500$, corresponding to the relevant boxplots in Fig. 4.12 (Model **M3**).

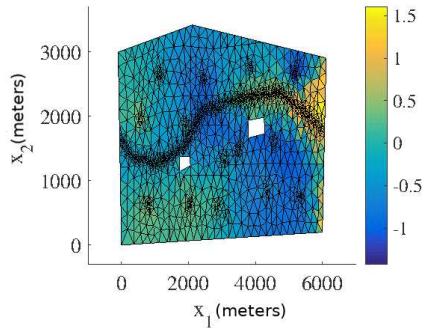
maintained at $h = -20.7$ cm ($\theta = 0.267$ cm^3 cm^{-3}), and along the bottom boundary $x_3 = 0$ cm, it is maintained at $h = -61.5$ cm. At all other boundaries a no-flow condition is imposed: $\nabla h \cdot \mathbf{n} = 0$, where $\mathbf{n}$ is the unit, outwardly pointing normal to the surface. The initial condition is $h(\mathbf{x}, 0) = -61.5$ cm.



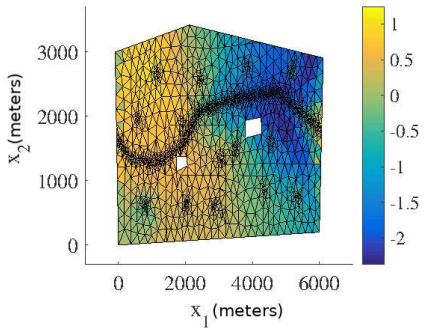
(a) True value, $N = 300$.



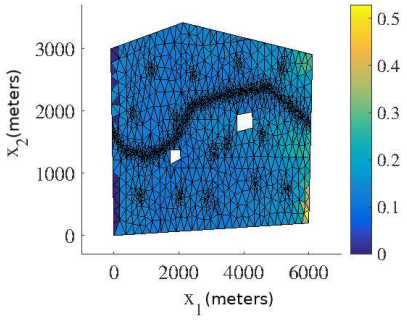
(b) True value, $N = 500$.



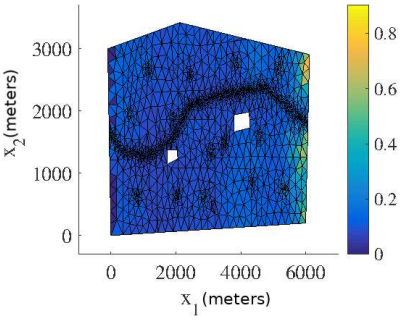
(c) Mean of the mean, $N = 300$.



(d) Mean of the mean, $N = 500$.



(e) Mean of the standard deviation, $N = 300$.

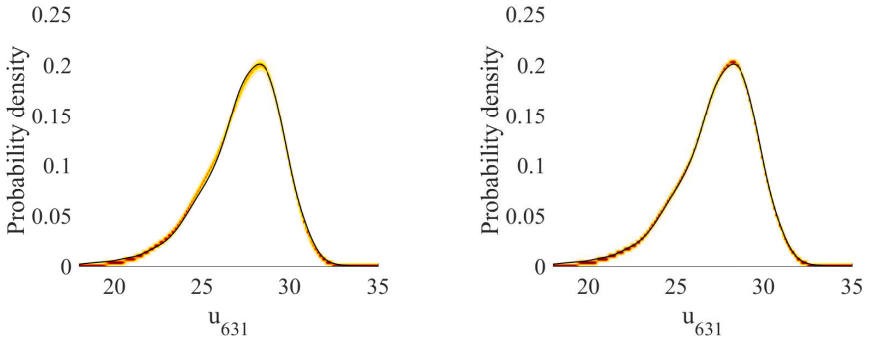


(f) Mean of the standard deviation, $N = 500$.

Figure 4.14: The test predictive means and standard deviations for predictions of the normalised pressure head with errors at the median from emulators using $P = 70$, $N = 300$ and with $P = 80$, $N = 500$, corresponding to the relevant boxplots in Fig. 4.12 (Model **M3**).

The covariance function for the random field $Z(\mathbf{x})$ is again of the form:

$$c_Z(\mathbf{x}, \mathbf{x}') = \sigma_Z^2 \exp\left\{ -\frac{(x_1 - x_1')^2}{l_1^2} - \frac{(x_2 - x_2')^2}{l_2^2} - \frac{(x_3 - x_3')^2}{l_3^2} \right\}, \tag{4.40}$$



(a) 300 training points, $P = 70$. (b) 500 training points, $P = 80$.

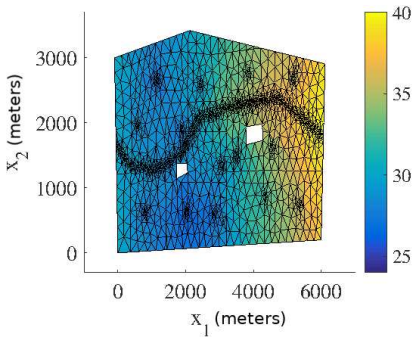
Figure 4.15: The pdfs of the pressure head response at the spatial coordinate $\mathbf{x} = \mathbf{x}_{631} = (2511, 486)$ on the finite-difference grid, obtained using kernel density estimation on $Q = 5000$ points (Model **M3**). The black line gives the MC prediction using the simulator. The contour shows how the emulator predictions vary with hyperparameter, precision and predictive distribution samples.

in which the l_i are correlation lengths, chosen as $l_1 = l_2 = l_3 = 7.5$ cm. The mean m_Z and variance σ_Z^2 are chosen such that the mean and standard deviation of $K(\mathbf{x})$ are 0.0094 cm s^{-1} (Shahraiyni and Ataie-Ashtiani [2011], Haverkamp et al. [1977]) and 0.00235 cm s^{-1} (25 % of the mean), respectively. The generalised variance satisfies $\sum_{i=1}^{k_\xi} \sqrt{\lambda_i} / \sum_{i=1}^{n} \sqrt{\lambda_i} = 0.75$ for $k_\xi = 15$.

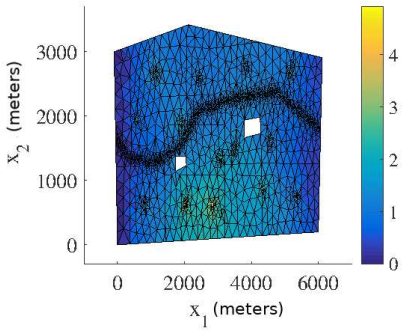
The training and test input samples were drawn independently: $\boldsymbol{\xi}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\boldsymbol{\xi}_q \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ to yield $\{\mathbf{y}_n\}_{n=1}^N$ for training and $\{\widetilde{\mathbf{y}}_q^*\}_{q=1}^Q$ for testing and UQ. We set $Q = 5000$ and $N \leq 800$. As before, the manifold dimension was set to $k_z = k_\xi$. The number of neighbours P and the number of training points N were chosen as in the first example by examining the errors $e_q = ||\widetilde{\mathbf{y}}_q^* - \overline{\mathbf{y}}_q^*|| / ||\widetilde{\mathbf{y}}_q^*||$ on the test set, where again $\widetilde{\mathbf{y}}_q^*$ is the solver output (truth) and $\overline{\mathbf{y}}_q^*$ is emulator prediction based on the GP predictive mean (4.18).

Equation (4.38) was solved using a finite difference scheme with first order differencing for the first order derivatives, central differencing for the second order derivatives and a fully implicit backward Euler time stepping scheme. A picard iteration scheme is used (Celia et al. [1990]) at each time step. Details are provided in Appendix 4C.

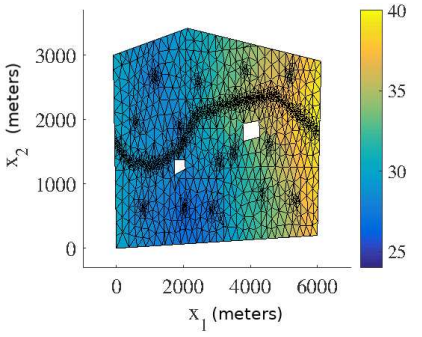
We followed the procedure of the first example. Training point numbers below 600 led to inaccurate results. For $N = 600$, the results were reasonably accurate but to achieve good accuracy we required $N > 700$. We present the results for $N = 800$. The pressure head is normalised as in the first example in order to



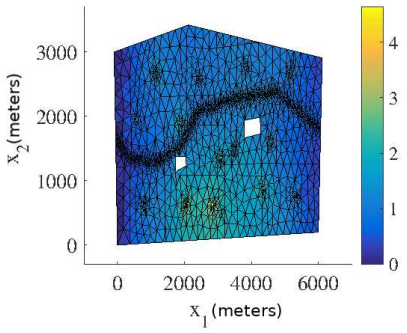
(a) Monte Carlo mean.



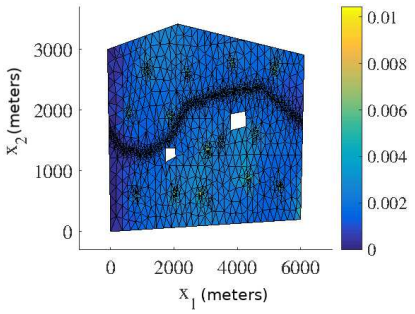
(b) Monte Carlo standard deviation.



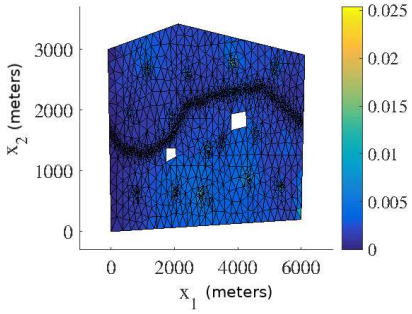
(c) Mean of the mean for 500 training points.



(d) Mean of the standard deviation for 500 training points.



(e) Standard deviation of the mean for 500 training points.

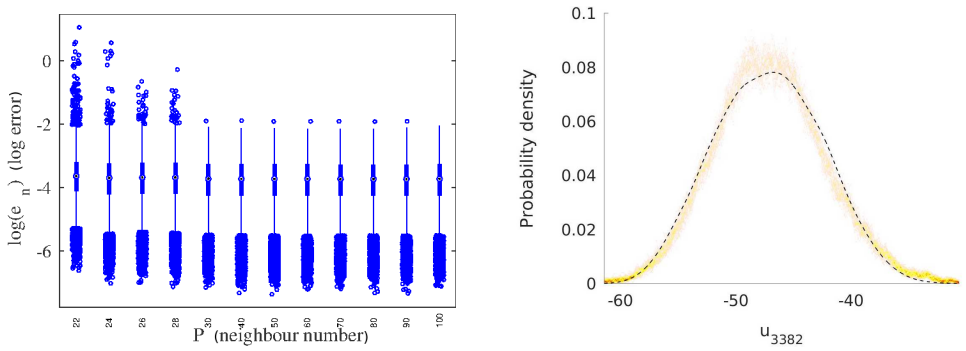


(f) Standard deviation of the standard deviation for 500 training points.

Figure 4.16: Moments of the mean and standard deviation for $P = 80$, $N = 500$ of the pressure head (Model **M3**). The emulator variation is a consequence of the hyperparameter and predictive distribution samples. We have a single, parametrised realisation of the manifold.

highlight the errors in the predictions more clearly. In Fig. 4.17(a) we plot the log normalised error $\ln(e_q)$ for an emulator trained on $N = 800$ points $\mathbf{y}_n$ and tested with $Q = 5000$ points $\widetilde{\mathbf{y}}_q^*$ for different nearest neighbour numbers $P > 20$ (averaging

over hyperparameter and precision posterior samples). For $P \leq 20$ the errors were high, with the same trend as seen in the first example.



(a) 800 training points.

(b) 800 training points, 30 k-NN.

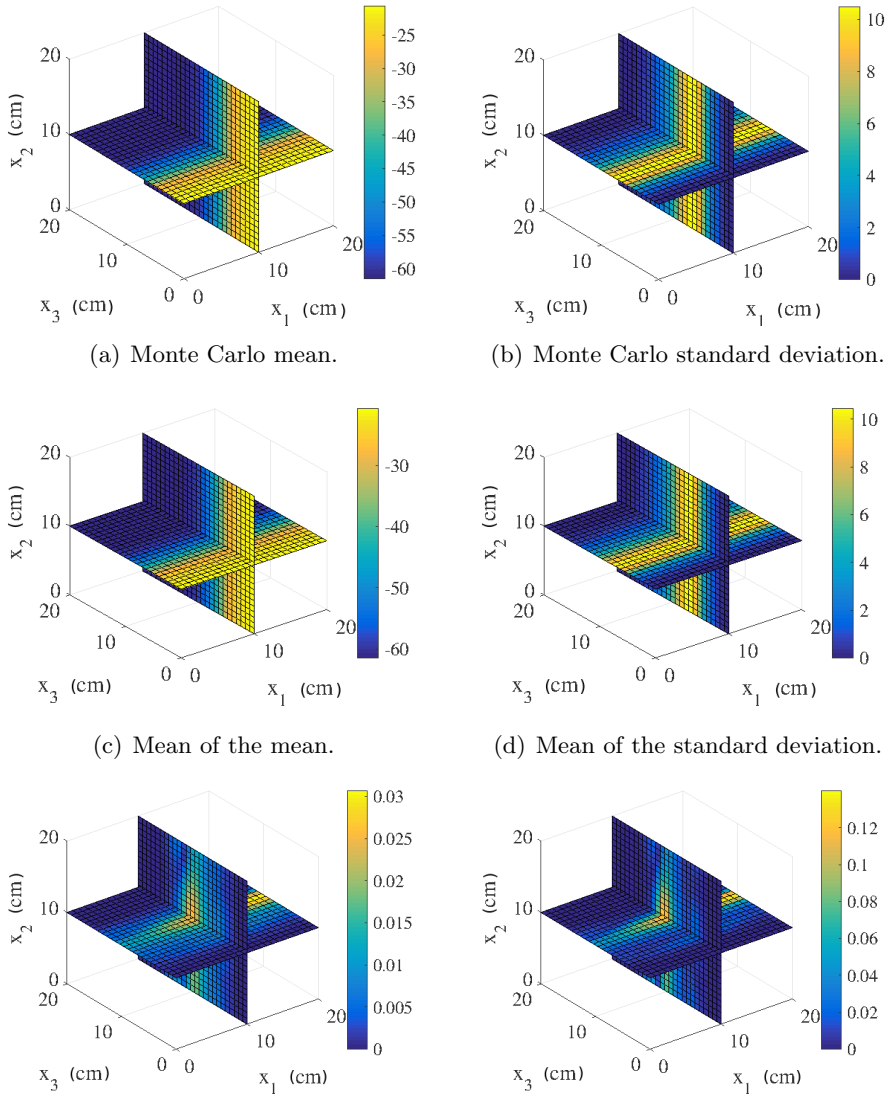
Figure 4.17: (a) Log normalised error $\ln(e_q)$ for an emulator trained on $N = 800$ points $\mathbf{y}_n$ and tested with $Q = 5000$ test points $\widetilde{\mathbf{y}}_q^*$ for different nearest neighbour numbers P. Predictions were obtained by averaging over hyperparameter and precision posterior samples. (b) The pdfs of the pressure head response at the location $\mathbf{x} = (10.4, 10.4, 10.4)^T$ ($N = 800$), obtained using kernel density estimation on $\{\widetilde{\mathbf{y}}_q^*\}_{q=1}^Q$. The black line gives the MC prediction using the simulator.

We use Algorithm 8 and KDE to obtain predictions of the pdf of a feature of the response. We choose as a feature the pressure head at the location $\mathbf{x} = (10.4, 10.4, 10.4)^T$ (grid point number 4411). The distributions are shown in Fig. 4.17(b) for $N = 800$. We can again find the means and standard deviations across predictive posterior, hyperparameter and precision samples to obtain distributions over the moments of the marginalised distribution (4.22). These are plotted in Fig. 4.18, alongside comparisons to the true values obtained from $\{\widetilde{\mathbf{y}}_q^*\}_{q=1}^Q$. These results show that the emulator performs extremely well, accurately capturing both the mean and standard deviation with high precision.

4.5 Numerical computation

LTSA naturally lends itself to parallel processing since almost all computations are performed on each neighbourhood independently. After merging threads we need to solve an eigenvalue problem for an $N \times N$ matrix. Similarly, independent Gaussian processes across latent dimensions leads to a natural parallel framework.

For large sample sizes and feature space dimensions storing each Q_i in memory can become infeasible ($N \times k_y \times k_z$ elements). Similarly, for large sample and neighbourhood sizes saving f can become infeasible ($N \times k^2$ elements). In such



(a) Monte Carlo mean.

(b) Monte Carlo standard deviation.

(c) Mean of the mean.

(d) Mean of the standard deviation.

(e) Standard deviation of the mean.

(f) Standard deviation of standard deviation.

Figure 4.18: Moments of the mean and standard deviation of the pressure head for $P = 30$, $N = 800$. Shown are the planes $x_1 = 10.4$ cm and $x_2 = 10.4$ cm. The emulator variation is a consequence of the hyperparameter and predictive distribution samples. We have a single, parametrised realisation of the manifold.

cases, these tensors may be saved to file or re-calculated online.

The scalability of our approach is limited by the computational complexity of Gaussian processes $\mathcal{O}\left(N^3\right)$. However, this can be alleviated by using sparse Gaussian process regression models. These models introduce m inducing points, reducing computational complexity to $\mathcal{O}\left(m^2 N\right)$. We may also use active learning

or a space filling design to reduce the number of samples required.

4.6 Discussion

We use nonparametric manifold learning in the form of LTSA (Zhang and Zha [2004]) to perform Bayesian inference (GP regression/emulation with Markov chain Monte Carlo) in an abstract feature space, and use an inverse (pre-image) map to obtain the output field at a finite number of points for an arbitrary input. We note that this method is not fully Bayesian as LTSA relies on a number of nearest neighbours hyperparameter which we do not place a prior on. In contrast to diffusion maps, Isomaps and kPCA, LTSA is a local method in that it approximates points on the manifold on localised regions (patches), rather than directly seeking a global basis for the feature space. This can potentially provide more accurate results, although this is of course dependent upon the sampling methodology for the points and the quality of the reconstruction mapping.

The aforementioned approach is combined with a Karhunen-Loève expansion for a log-normally distributed input field and a framework for UQ is developed. We derive analytical forms for the output distribution by pushing the feature-space Gaussian distribution through a locally linear reconstruction map. Additionally, we derive analytical estimates of the moments of the predictive distribution *via* approximate marginalization of the stochastic input. To sample from the hyper-parameter and signal precision posteriors we employ a HMC scheme and use MC sampling to approximately marginalize the stochastic input distribution. The accuracy of the approach is demonstrated *via* two examples: a linear, steady state Darcy's Law with a contaminant mass balance in a 2-d domain (aquifer) and a time-dependent Richards equation evaluated at a fixed time in a 3-d domain. In both cases we consider a stochastic hydraulic conductivity input.

In this chapter we developed a new approach to the emulation of a model involving a random field input and a field output, with a focus on problems arising in groundwater flow modelling. The main challenges are the high input and output space dimensionality, which we dealt with using a KLE and manifold learning, respectively. We implemented LTSA on the given outputs (training data), which allowed us to perform Bayesian inference in a low dimensional feature space. Furthermore, we developed a framework for UQ in such problems by marginalising over the inputs, either analytically (the mean and possibly in some cases the standard deviation) or using MC sampling.

Testing the emulation method on two examples reveals that it performs well

in certain cases. When the variance of the log-normal input is high or the correlation lengths of the normal process $Z(\mathbf{x})$ are short, the accuracy suffers, as is found in all other approaches. Nevertheless, the accuracy in terms of the forward UQ problem is high even in such cases for the examples considered (of course, further increases in the variance and correlation lengths would eventually lead to unacceptably poor performance).

The major drawback of the KLE approach (and similarly with circulant embedding) is the curse of dimensionality as the number of retained coefficients grows. Some progress can be made in this regard by using a Smolyak algorithm (Smolyak [1963]) for sampling or incremental local tangent space alignment (Liu et al. [2006]) combined with active learning (Settles [2012]), but the gains will be limited. Given current computational resources, our method (in common with other methods except direct Monte Carlo or ROMs) is, therefore, potentially limited to problems in which the domain size is at most a few multiples of the shortest correlation length. The assumption of independence of the feature vector coordinates is also sub-optimal. Since the number of coordinates is small, however, this assumption can easily be relaxed by adopting, e.g., a convolved GP approach.

Chapter 5

Enriched mixtures of generalised Gaussian process experts

The standard Gaussian process regression model has been successfully applied to various problems, for example, in geo-statistics (Matheron [1973]), atmospheric sciences (Daley [1991]) and medicine (Myllymäki et al. [2014]), to name a few; however, it is limited in the sense that it only allows for flexibility in the regression function. Many datasets require further flexibility in the error distribution and present departures from the model, such as non-normal or multi-modal errors and different error variances, degrees of skewness or tail behaviour in different regions of the input space. Moreover, for computational purposes, a stationary assumption of the GP is typically employed. This is inappropriate in many practical examples as it limits the model's ability to recover changing behaviour of the regression function across the input space, such as different levels of smoothness, variability or periodicity in different regions of the input space.

Mixtures of experts (Jacobs et al. [1991]) provide an approach to address these issues by partitioning the input space. Each expert is a conditional model for y given x, and a gating network is used to map experts to local regions of the input space. The scalability of the method is enhanced since each expert may solely consider its local region and simplifying assumptions such as stationarity and continuity of the regression function need only hold locally in each region. In Tresp [2001], mixtures of GP experts were introduced to increase model flexibility and recover local non-linear regression functions. However, the number of mixture components plays a key role in the flexibility of the model. In practice, this is often chosen through

post-processing techniques, with sparse or over-fitted mixtures (Malsiner-Walli et al. [2016]), or with a Bayesian prior on the number of components. The latter requiring posterior sampling through, for example, reversible jump Markov chain Monte Carlo methods (Green [1995]).

Infinite mixtures of GP experts were introduced in Rasmussen and Ghahramani [2002]. This Bayesian nonparametric approach is highly flexible, allowing the data to determine the number of clusters present. Moreover, this number can grow unboundedly as more and more data points are observed. An alternative infinite mixture of GP experts was introduced in Meeds and Osindero [2006], where the joint distribution of the inputs and targets is explicitly modelled, while in Rasmussen and Ghahramani [2002] only the conditional distribution of y given x is modelled. Advantages of modelling the joint distribution include the ability to handle missing data and answer inverse problems, as well as simplified computations that rely on established algorithms of infinite mixtures for exchangeable data (Neal [2000]).

In Meeds and Osindero [2006], a multivariate Gaussian distribution for the inputs is used for each cluster. When the marginal distribution of the inputs is complex, this can lead to an unnecessary number of experts being created, degrading the predictive performance and increasing uncertainty. In Yuan and Neubauer [2009], this constraint is removed by using a Gaussian mixture model for the input density of each expert. However, a finite dimensional approximation of the infinite mixture is used at both levels. Moreover, the multivariate Gaussian local input model scales poorly with the input dimension P due to the computational cost of dealing with the full P by P matrix.

The Treed-GP (Gramacy and Lee [2008]) is another example of a mixture of GP experts. The input space is partitioned with axis-aligned rectangular partitions and a MCMC inference approach is used to sample over the space of partitions. However, this axis-aligned approach also scales poorly with increasing input dimension, again leading to an unnecessarily large number of experts. More flexible partitioning approaches such as Voronoi tessellations (Pope et al. [2018]) have been proposed, where the partition boundaries are no longer axis-aligned. However, inference becomes computationally more expensive, especially as the number of input dimensions increases.

The problems associated an unnecessary large number of experts was highlighted in Wade et al. [2014] for linear regression experts, where they demonstrated a loss of predictive accuracy and increased uncertainty, particularly, as the dimensionality of the input space increases. Due to the greater flexibility of GP experts over linear regression experts, these problems are exacerbated for mixtures of GP

experts. Following Wade et al. [2014], we propose to overcome this by constructing a mixture of GP experts based on the enriched Dirichlet process (EDP) (Wade et al. [2011]), which allows for a nested partitioning scheme. Moreover, we make local independence assumptions of the inputs, which are necessary to ease computations, particularly as the dimension of the input space increases, and to allow for inclusion of multiple types of inputs. This construction allows our model to scale to higher dimensional input spaces without sacrificing parsimony and predictive accuracy, while also maintaining a simple analytically computable allocation rule for efficient MCMC inference.

Infinite mixtures of generalised linear experts were introduced in Hannah et al. [2011] to provide a unifying framework for mixtures of linear experts to model multiple response types. Building on this, we propose an infinite mixture of generalised GP experts, combining generalised Gaussian process models (GGPM) (Chan and Dong [2011]) to specify non-linear experts for multiple response types with the flexibility and scalability of mixtures.

We developed an efficient MCMC algorithm based on non-conjugate collapsed Gibbs sampling. The advantages of the proposed model and algorithm are demonstrated on a highly non-linear toy example, with increasing input dimension, and an Alzheimer's Disease Neuroimaging Initiative challenge to predict decline in cognitive impairment, with ordinal response.

The chapter is organised as follows. In section 5.1, we introduce the joint mixture of generalised Gaussian process experts, extending Meeds and Osindero [2006] for multiple response and input types. A Gibbs sampling algorithm for posterior inference is described in section 5.1.1, and in section 5.1.2, we detail how to make predictions and summarise the clustering structure from MCMC output. The enriched mixture of generalised Gaussian process experts is proposed in section 5.2, with extended Gibbs sampling algorithm in section 5.2.1 and predictions and clustering tools provided in section 5.2.2. Examples are provided in section 5.3, with a highly non-linear toy example with increasing input dimension in section 5.3.1, and an Alzheimer's disease example with ordinal response and multiple input types in section 5.3.2.

5.1 Joint mixture of generalised Gaussian process experts

A mixture model for the joint density of the output y and the P-dimensional input x assumes

$$f(y,x) = \int p(y|x,\theta)p(x|\psi)dQ(\theta,\psi), \tag{5.1}$$

where $p(y|x,\theta)$ for $\theta \in \Theta$ is a family of densities on output space, $p(x|\psi)$ for $\psi \in \Psi$ is a family of densities on input space, and the unknown mixing measure Q is a probability measure on $\Theta \times \Psi$. A Bayesian nonparametric approach places a prior on the infinite dimensional parameter Q, and the Dirichlet process (Ferguson [1973b]), denoted $Q \sim \mathrm{DP}(\alpha Q_0)$, is the most popular choice, due to its large support, analytic tractability, and interpretable hyperparameters. It is characterised by its finite dimensionals; for any measurable partition $B_1, \ldots, B_k$ of $\Theta \times \Psi$ with $k \in \mathbb{N}$,

$$(Q(B_1), \ldots, Q(B_k)) \sim \mathrm{Dir}(\alpha Q_0(B_1), \ldots, \alpha Q_0(B_k)),$$

where $\alpha > 0$ and Q_0 is a probability measure on $\Theta \times \Psi$. The two parameters of the DP, α and Q_0, are easy to interpret and elicit; Q_0 is the prior guess of Q, and α is the precision parameter, controlling the strength of belief in Q_0.

The DP is discrete with probability one, and an equivalent formulation of the DP mixture model for the joint density in (5.1) is based on the stick-breaking construction [Sethuraman, 1994a] of the DP;

$$f(y,x) = \sum_{j=1}^{\infty} w_j p(y|x,\theta_j)p(x|\psi_j), \quad \text{where} \quad w_j = v_j \prod_{j'<j}(1-v_{j'}), \quad v_j \overset{iid}{\sim} \mathrm{Beta}(1,\alpha), \tag{5.2}$$

and (θ_j), (ψ_j), and (w_j) are independent with $(\theta_j, \psi_j) \overset{iid}{\sim} Q_0$. For data points (y_n, x_n), $n = 1, \ldots, N$ conditionally independent and identically distributed from (5.2), the countably infinite number of mixture components induces a random partition of the N data points into clusters, where the number of clusters can grow unboundedly with the data. Introducing the latent variables z_n denoting the cluster allocation of data point n, in order of appearance, and the parameters (θ_j, ψ_j) denoting the parameters of the j^{th} observed cluster, the unknown mixing measure

Q can be marginalised. In this case, the model can be expressed as

$$y_n|x_n, z_n = j, \theta_j \overset{ind}{\sim} p(y_n|x_n, \theta_j), \quad x_n|z_n = j, \psi_j \overset{ind}{\sim} p(x_n|\psi_j), \tag{5.3}$$

where $(\theta_j, \psi_j) \overset{iid}{\sim} Q_0$. The law of allocation variables (z_n) is defined by the sequence of predictive distributions (Blackwell and MacQueen [1973])

$$z_{N+1}|z_1, \ldots, z_N \sim \frac{\alpha}{\alpha + N}\delta_{k_N+1} + \sum_{j=1}^{k_N} \frac{N_{j,N}}{\alpha + N}\delta_j,$$

where k_N denotes the number of clusters in the sample of size N and $N_{j,N}$ denotes the number of data points allocated to cluster j in the sample of size N (when the sample size is clear, we will drop the subscript notation N). The mass parameter α of the DP strongly influences the number of clusters, and we will consider a prior on this parameter as well, $\alpha \sim \text{Ga}(u_\alpha, v_\alpha)$.

We define the local expert $p(y|x, \theta)$ to be an extension of the generalised linear model (GLM) used in Hannah et al. [2011]. Specifically, the local distribution of y belongs to the exponential family, which in canonical form assumes

$$p(y|x, \theta) = \exp\left(\frac{y\eta - b(\eta)}{a(\phi)} + c(y, \phi)\right).$$

The functions a, b, and c are known and specific to the exponential family; ϕ is the scale parameter; and η is the canonical parameter with $b'(\eta) = \mu(x) = \mathbb{E}[y|x]$ and $g(\mu(x)) = m(x)$, where g is a chosen link function that maps $\mu(x)$ to the real line. In GLMs (McCullagh and Nelder [1989]), a linear function of x determines the canonical parameter through a set of transformations, that is $m(x) = \tilde{x}\beta$ (with $\tilde{x} = [1, x^T]$). We extend this to allow for a general non-linear function and place a GP prior on the unknown function. The parameters of this generalised Gaussian process model (GGPM) (Chan and Dong [2011]) for mixture component j are $\theta_j = (m_j, \beta_{0,j}, \lambda_j, \phi_j)$ with priors

$$m_j|\beta_{0,j}, \lambda_j \overset{ind}{\sim} \text{GP}(\beta_{0,j}, K_{\lambda_j}), \quad \lambda_j \overset{iid}{\sim} \pi(\lambda), \quad \beta_{0,j} \overset{iid}{\sim} \pi(\beta_0), \quad \text{and} \quad \phi_j \overset{iid}{\sim} \pi(\phi).$$

Here, $m(x) \sim \text{GP}(\beta_0, K_\lambda)$ denotes a Gaussian process prior on the unknown function $m(x)$, with constant mean function, i.e. $\mathbb{E}[m(x)] = \beta_0$, and kernel function K_λ with hyperparameters λ, defining the covariance of the function at any two inputs, $\text{Cov}(m(x), m(x')] = K_\lambda(x, x')$. We note that in typical GP regression models, it is common to use a zero centred GP prior, which is made appropriate by subtracting

the overall mean from the response. However, in this case, we cannot centre the data within component, as the clustering structure is unknown; thus, it is appropriate to include a constant mean function. Additionally, due to the importance of the parameters of GP kernel, we assume that these parameters λ_j are component specific.

A list of examples of generalised Gaussian process experts is provided in appendix C.1. Three examples considered in section 5.3 include 1) the **Gaussian** with identity link function,

$$p(y|x, \theta_j) = \mathrm{N}(y|m_j(x), \sigma_j^2);$$

2) the **Bernoulli** with probit link,

$$p(y|x, \theta_j) = \mathrm{Bern}(y|\Phi(m_j(x)),$$

where Φ is the standard normal cumulative distribution function; and 3) the **ordinal** with probit link and ordered categories $l = 0, \ldots, L$ and cutoffs $0 = \varepsilon_0 < \varepsilon_1 < \ldots < \varepsilon_{L-1}$,

$$\mathbb{P}(y \leq l|x, \theta_j) = \Phi\left(\frac{\varepsilon_l - m_j(x)}{\sigma_j}\right).$$

Due to the nonparametric nature of the model, we consider fixed cutoffs $\varepsilon_1, \ldots, \varepsilon_{L-1}$. With this choice of link function, the model can be equivalently formulated through a latent Gaussian response $\tilde{y}$;

$$\tilde{y} \sim \mathrm{N}(m_j(x), \sigma_j^2) \quad \text{and} \quad p(y|x, \theta_j) = \begin{cases} \mathbf{1}(\tilde{y} \leq 0) & \text{if } l = 0 \\ \mathbf{1}(\varepsilon_{l-1} < \tilde{y} \leq \varepsilon_l) & \text{if } l = 1, \ldots, L-1 \\ \mathbf{1}(\tilde{y} > \varepsilon_{L-1}) & \text{if } l = L \end{cases},$$

with the ordered probit model recovered by marginalising the latent $\tilde{y}$.

The model $p(x|\psi)$ assumes local independence of the P-dimensional input where each local model belongs to the exponential family, that is,

$$p(x|\psi) = \prod_{p=1}^{P} p(x_p|\psi_p) = \prod_{p=1}^{P} \exp(\psi_p' t_p(x_p) - a_p(\psi_p) + b_p(x_p)). \tag{5.4}$$

The parameter ψ has the standard conjugate prior, which assumes independence of ψ_p across $p = 1, \ldots, P$ with

$$\pi(\psi_p) = \exp(\psi_p' \tau_p - \nu_p a_p(\psi_p) + c_p(\tau_p, \nu_p)).$$

In this conjugate setting, the parameters ψ can be marginalised and the marginal and predictive likelihood of the inputs in each cluster are available analytically. Some examples of local input models are provided in appendix C.2.

The DP mixture of GP expert models proposed in Meeds and Osindero [2006],Yuan and Neubauer [2009] and Nguyen and Bonilla [2014] are based on the case of continuous inputs and outputs, i.e., the Gaussian likelihood with identity link is used for y. The authors further utilise a multivariate Gaussian density for $p(x|\psi)$ and chose the conjugate normal-inverse Wishart as the prior for covariance matrix in the multivariate Gaussian. However, for even moderately large P, this approach is practically unfeasible. Indeed, the computational cost of dealing with the full P by P covariance matrix greatly increases with large P. Furthermore, the conjugate inverse Wishart prior is known to be too poorly parametrised; in particular, there is a single parameter to control variability, regardless of P (Consonni and Veronese [2001]).

Our model differs from previous proposals by 1) generalising for other types of outputs using a GLM framework and 2) assuming local independence of the inputs in equation (C.1). Computationally, reducing the covariance matrix to P variances can greatly ease calculations. Concerning flexibility of the base measure, the conjugate prior now includes a separate parameter to control variability for each of the P variances. Also, the assumption of local independence of the inputs allows for easy inclusion of discrete or other types of inputs. Note that even though, within each component, we assume independence of the covariates, globally, there may be dependence.

5.1.1 Non-conjugate collapsed Gibbs sampling

For inference with the joint mixture of generalised GP experts, we resort to Markov chain Monte Carlo (MCMC) algorithms. Specifically, we consider a collapsed Gibbs sampler, which is based on the model formulation in (5.3) that marginalises over the mixing measure Q by parametrising in terms of the latent allocation variables $z_{1:N}$ and unique cluster parameters $(\theta_j, \psi_j)_{j=1}^k$. Additionally, as we make use of the standard conjugate priors for ψ_j, these parameters can also be marginalised.

In general the parameters θ_j cannot be marginalised. In the following, we consider the case when the functions m_j may be marginalised. This includes the Gaussian likelihood with identity link, but also the probit model for binary outputs, the ordered probit model for ordinal outputs, and the multinomial probit model for categorical outputs, through data augmentation techniques. In the latter, the data is augmented with latent Gaussian outputs $\tilde{y}_{1:N}$, which have a deterministic

relationship with the observed outputs. In the general case, when the functions m_j cannot be marginalised, strong dependence will exist in a Gibbs sampling algorithm, which alternatively samples the functions m_j and the kernel hyperparameters λ_j, resulting in poor mixing. To overcome this, re-parametrisations (Yu and Meng [2011]) or a pseudo-marginal algorithm (Filippone and Girolami [2014]) may be considered. The MCMC gives posterior samples

$$\zeta_m = (z_{1:N}^{(m)}, \sigma_{1:k^{(m)}}^{2\,(m)}, \beta_{0,1:k^{(m)}}^{(m)}, \lambda_{1:k^{(m)}}^{(m)}, \alpha^{(m)}, \tilde{y}_{1:N}^{(m)}) \quad \text{for } m = 1, \ldots, M,$$

from the posterior

$$\pi(z_{1:N}, \sigma_{1:k}^2, \beta_{0,1:k}, \lambda_{1:k}, \alpha, \tilde{y}_{1:N} \mid y_{1:N}, x_{1:N}),$$

through a Gibbs sampling algorithm, which alternatively samples each parameter from its full conditional. The allocation variables $z_{1:N}$ are sampled with a collapsed Gibbs sampler, combining Algorithm 3 for the conjugate parameters that can be marginalised and Algorithm 8 for the non-conjugate parameters that cannot be marginalised of Neal [2000]. The unique cluster parameters $(\sigma_j^2, \beta_{0,j}, \lambda_j)$ are conditionally independent across $j = 1, \ldots, k$ and updated with a HMC scheme (Duane et al. [1987]). The mass parameter α is updated using the auxiliary variable technique of Escobar and West [1995]. Finally, the latent outputs $\tilde{y}_{1:N}$ (if present) are sampled from truncated multivariate Gaussians through Gibbs sampling and cumulative distribution function inversion techniques. A full description of the algorithm is provided in appendix C.3.

In the collapsed Gibbs sampler, N steps are performed to update each allocation variable z_n conditioned on all others $z_1, \ldots, z_{n-1}, z_{n+1}, \ldots, z_N$. In the examples considered, this resulted in sufficient mixing, however, in some problems split-merge updates (Jain and Neal [2004, 2007]) may be needed to allow for global changes to the allocation variables.

5.1.2 Predictions and clustering

Predictions. Given the MCMC samples, we can compute predictions for the new output y_* given x_*. For example, in the Gaussian case, we may interested in the posterior expectation of y_*, which is the prediction of y_* under the squared error

loss function. This is given by

$$\mathbb{E}[y_*|x_*, y_{1:N}, x_{1:N}] = \int \mathbb{E}[y_*|x_*, y_{1:N}, x_{1:N}, \zeta] \frac{\pi(\zeta|y_{1:N}, x_{1:N}) f(x_*|x_{1:N}, \zeta)}{f(x_*|x_{1:N})} d\zeta$$
$$\approx C^{-1} \left(\sum_{m=1}^{M} p^{(m)}_{k^{(m)}+1}(x_*)\mu_\beta + \sum_{j=1}^{k^{(m)}} p^{(m)}_j(x_*)\widehat{m}^{(m)}_j(x_*) \right),$$

$\widehat{m}^{(m)}_j(x_*)$ denotes the GP predictive mean in cluster j for sample m with

$$p^{(m)}_{k^{(m)}+1}(x_*) = \frac{\alpha^{(m)}}{\alpha^{(m)} + N} h(x_*), \quad p^{(m)}_j(x_*) = \frac{N^{(m)}_j}{\alpha^{(m)} + N} h(x_*|\mathbf{X}^{(m)}_j),$$

and $C = \sum_{m=1}^{M} p^{(m)}_{k^{(m)}+1}(x_*) + \sum_{j=1}^{k^{(m)}} p^{(m)}_j(x_*)$. Here, $h(x_*)$ denotes the marginal density of x^* and $h(x_*|\mathbf{X}_j)$ denotes the predictive marginal density of x_* given $\mathbf{X}_j$, which contains the x_n such that $z_n = j$ (see appendix C.2 for more details and examples). Thus, the posterior expectation is a weighted average of the GP predictions for each cluster with weight proportional to the number of points in that cluster times the similarity between the new input and the inputs in that cluster, as measured by the local predictive marginal likelihood for the inputs, and the prediction from a new cluster (with μ_β denoting the expectation of β_0 under the prior) with weight proportional to α times the local marginal likelihood for x_*. Similarly, we can compute the predictive density for a new output y_* or appropriate predictive quantities for other types of outputs; details are provided in appendix C.4. An advantage of jointly modelling the outputs and inputs includes the possibility to compute predictive quantities of y_* based only on a subset of inputs, by marginalising over the other inputs. This is further detailed in appendix C.4.

Clustering. The joint infinite mixture of experts induces a latent clustering of data points into groups. In many cases, it may be of interest to examine this latent clustering to identify groups of data points with similar inputs and similar relationship between the inputs and outputs and more generally, to improve understanding of the model. The model provides a posterior over the clustering structure, and the MCMC algorithm gives samples from this posterior. To summarise the MCMC samples of clusterings and obtain a point estimate of the clustering structure, we consider the estimate $\widehat{z}_{1:N}$ based on minimising the posterior expected variation of information (Wade and Ghahramani [2017]) with accompanying **R** package (Wade [2015]). Based on this clustering estimate, we can compute the (marginal) posterior allocation probabilities to each cluster for a new data point with input x_* and the

GP predictive means for each cluster; this quantity may depend on hyperparameters, e.g. the mass parameter α for allocation probabilities, and in this case, we may plug-in a point estimate of the hyperparameters, e.g. the MAP estimate of $\widehat{\alpha}$ that optimises

$$(u_\alpha + \widehat{k} - 1)\log(\alpha) - v_\alpha \alpha - \sum_{j=1}^{N} \log(\alpha + j - 1).$$

From the MCMC samples, we can also compute the posterior similarity matrix with entries

$$p(z_n = z_{n'} | y_{1:N}, x_{1:N}) \approx \frac{1}{M} \sum_{m=1}^{M} \mathbf{1}(z_n^{(m)} = z_{n'}^{(m)}),$$

representing the posterior probability that two data points are clustered together. This matrix provides an understanding of the uncertainty in the clustering structure.

5.2 Enriched mixture of generalised Gaussian process experts

The joint mixture of generalised GP experts allocates data points to groups in order to obtain a good approximation to the joint density of the inputs and outputs. This means that data points with similar x and similar relationship between y and x tend to cluster together. The assumption of local independence of the inputs together with the assumption that the local input model belongs to the exponential family are crucial for scaling the model to higher dimensional input spaces and for inclusion of multiple input types. However, these assumptions result in a rigid similarity measure between inputs, that can cause the posterior to concentrate on partitions with many small clusters, as needed to describe the marginal of x. This is particularly true when the marginal of x is complex and the inputs are highly dependent, and as P, the dimension of the inputs, increases. This occurs despite the flexible nature of the GP model, typically requiring few GP experts to approximate the conditional of y given x, and results in degradation of regression and conditional density estimates with wide credible intervals due to the unnecessarily small sample sizes for each GP expert.

To overcome this, we replace the DP prior on the mixing measure with the enriched Dirichlet process (EDP) (Wade et al. [2011]) that allows nested clustering of x within each cluster of y and maintains a simple analytically computable allocation rule. Wade et al. [2014] demonstrated the advantages of this approach when experts were simple linear regression models. Here we achieve greater improvements due to

the flexibility of the GP experts in recovering the conditional density of y given x with fewer experts.

The EDP defines a prior on the unknown joint mixing measure Q of (θ, ψ) by expressing the joint probability measure in terms of the marginal and conditionals. This requires an ordering of θ and ψ, and to achieve the desired clustering structure, we consider the random marginal Q_θ and the random conditionals $Q_{\psi|\theta}(\cdot|\theta)$. The parameters of the EDP consist of a base measure Q_0 on $\Theta \times \Psi$; a mass parameter α_θ associated to θ; and a collection of mass parameters $\alpha_\psi(\theta)$ associated to ψ for every $\theta \in \Theta$. The EDP is defined by

$$Q_\theta \sim \mathrm{DP}(\alpha_\theta Q_{0\,\theta}), \quad \text{and} \quad Q_{\psi|\theta}(\cdot|\theta) \sim \mathrm{DP}(\alpha_\psi(\theta) Q_{0\,\psi|\theta}(\cdot|\theta)) \quad \text{for all } \theta \in \Theta,$$

and $Q_{\psi|\theta}(\cdot|\theta)$ are independent across $\theta \in \Theta$ and from Q_θ. Together these assumptions induce a prior for the joint mixing measure Q through the mapping $(Q_\theta, Q_{\psi|\theta}) \to \int Q_{\psi|\theta}(\cdot|\theta) dQ_\theta(\theta)$.

The enriched mixture of generalised Gaussian process experts assumes:

$$f(y, x) = \int p(y|x, \theta) p(x|\psi) dQ(\theta, \psi), \quad Q \sim \mathrm{EDP}(\alpha_\theta, \alpha_\psi(\theta), Q_0),$$

with local expert $p(y|x, \theta)$ and local input model $p(x|\psi)$ as specified in section 5.1. The model induces a nested clustering which partitions data points in y-clusters and x-subclusters with each y-cluster. The latent cluster allocation of each data point consists of two terms $z_n = (z_{y,n}, z_{x,n})$, where $z_{y,n} = j$ if the n^{th} data point belongs to j^{th} y-cluster with parameter θ_j and $z_{x,n} = l$ if the n^{th} data point belongs to l^{th} x-cluster with parameter $\psi_{l|j}$ within the j^{th} y-cluster. The random mixing measure can be marginalised, and the model can be equivalently expressed as

$$y_n | x_n, z_{y,n} = j, \theta_j \overset{ind}{\sim} p(y_n | x_n, \theta_j), \quad x_n | z_{y,n} = j, z_{x,n} = l, \psi_{l|j} \overset{ind}{\sim} \prod_{p=1}^{P} p(x_{n,p} | \psi_{l|j,p}).$$

The law of allocation variables ($z_n = (z_{y,n}, z_{x,n})$) is defined by the sequence of predictive distributions:

$$\begin{aligned}(z_{y,N+1}, z_{x,N+1}) | z_{1:N}, \theta_{1:k} \sim & \frac{\alpha_\theta}{\alpha_\theta + N} \delta_{k+1,1} + \sum_{j=1}^{k} \frac{N_j}{\alpha_\theta + N} \frac{\alpha_\psi(\theta_j)}{\alpha_\psi(\theta_j) + N_j} \delta_{j,k_j+1} \\ & + \sum_{j=1}^{k} \sum_{l=1}^{k_j} \frac{N_j}{\alpha_\theta + N} \frac{N_{l|j}}{\alpha_\psi(\theta_j) + N_j} \delta_{j,l},\end{aligned}$$

where k denotes the number of y-clusters of sizes N_j and k_j denotes the number x-clusters within the j^{th} y-cluster of sizes $N_{l|j}$. We further consider a prior on the mass parameters with $\alpha_\theta \sim \text{Ga}(u_\theta, v_\theta)$ and $\alpha_\psi(\theta)$ independent and identically distributed with $\alpha_\psi(\theta) \sim \text{Ga}(u_\psi, v_\psi)$ and make use of the short notation $\alpha_{\psi,j} = \alpha_\psi(\theta_j)$.

5.2.1 Non-conjugate collapsed Gibbs sampling

The non-conjugate collapsed Gibbs sampler is extended for the enriched mixture of generalised GP experts. This involves extending the collapsed Gibbs algorithm to sample the nested allocation variables $z_{1:N}$ and to improve mixing, includes a Metropolis-Hastings step which proposes to move an x-cluster to be nested within a new or different y-cluster. An additional step is included to update the mass parameters ($\alpha_{\psi,j}$). A full description of the algorithm is provided in appendix C.5.

5.2.2 Predictions and clustering

Predictions. Given the MCMC samples, prediction for a new output y_* given $x*$ can be computed similarly to section 5.1.2. Specifically, for the the Gaussian example, the posterior expectation of y_* is given by

$$\begin{aligned}\mathbb{E}[y_*|x_*, y_{1:N}, x_{1:N}] &= \int \mathbb{E}[y_*|x_*, y_{1:N}, x_{1:N}, \zeta] \frac{\pi(\zeta|y_{1:N}, x_{1:N}) f(x_*|x_{1:N}, \zeta)}{f(x_*|x_{1:N})} d\zeta \\ &\approx C^{-1} \left(\sum_{m=1}^{M} p^{(m)}_{k^{(m)}+1}(x_*)\mu_\beta + \sum_{j=1}^{k^{(m)}} p^{(m)}_j(x_*)\widehat{m}^{(m)}_j(x_*) \right).\end{aligned}$$

That is, for each posterior sample, the expectation of y_* is again a weighted average of the GP predictions from each cluster and from a new cluster. However, the input-dependent weights more flexibility measure the similarity between the new input and the inputs of each cluster, through a mixture model:

$$\begin{aligned}p^{(m)}_{k^{(m)}+1}(x_*) &= \frac{\alpha^{(m)}_\theta}{\alpha^{(m)}_\theta + N} h(x_*), \\ p^{(m)}_j(x_*) &= \frac{N^{(m)}_j}{\alpha^{(m)}_\theta + N} \left(\frac{\alpha^{(m)}_{\psi,j}}{\alpha^{(m)}_{\psi,j} + N^{(m)}_j} h(x_*) + \sum_{l=1}^{k^{(m)}_j} \frac{N^{(m)}_{l|j}}{\alpha^{(m)}_{\psi,j} + N^{(m)}_j} h(x_*|\boldsymbol{X}^{(m)}_{l|j}) \right),\end{aligned} \tag{5.5}$$

with $C = \sum_{m=1}^{M} p^{(m)}_{k^{(m)}+1}(x_*) + \sum_{j=1}^{k^{(m)}} p^{(m)}_j(x_*)$, and $\boldsymbol{X}_{l|j}$ containing the x_n such that $z_n = (j, l)$. Estimates of the predictive density or other appropriate predictive

quantities for other types of inputs can be computed as described in appendix C.6 with the more flexible input-dependent weights in (5.5).

Clustering. The enriched mixture of experts induces a latent nested clustering of data points into y-clusters and x-clusters within each y-cluster. To study the y-clustering at the first level, we utilise the tools described in section 5.1.2. Given an estimate of the y-clustering, we can further utilise the tools the described in section 5.1.2 to study the x-clustering within each estimated y-cluster.

5.3 Examples

We demonstrate the advantages of the enriched model in two examples. Specifically, these advantages over the joint model include improved predictive accuracy, smaller credible intervals while maintaining good coverage, and a more interpretable clustering structure. Additionally, we show the range of applicability of our model for continuous inputs and outputs, with increasing improvement over the joint model as P increases, in the first example; and for ordinal outputs with multiple input types in the second example. Code to reproduce the results is publicly available at GitLab/charles1992/MixtureOfExperts, alongside further plots and videos.

5.3.1 Simulated mixture of damped cosine functions

In the first example, a data set of 100 points was generated, where only the first input is a predictor for the output, as in Wade et al. [2014]. The true model for the output is a highly non-linear regression model obtained as a mixture of two non-linear damped cosine functions (Santner et al. [2003]), where the mixture weights only depend on the first input:

$$\begin{aligned} y_n|x_n \overset{ind}{\sim} & p\left(x_{n,1}\right) \mathrm{N}\left(\exp\left\{\beta_{1,0}x_{n,1}\right\}\cos\left(\beta_{1,1}\pi x_{n,1}\right), \sigma^2\right) \\ & + \left(1-p\left(x_{n,1}\right)\right)\mathrm{N}\left(\exp\left\{\beta_{2,0}x_{n,1}\right\}\cos\left(\beta_{2,1}\pi x_{n,1}\right), \sigma^2\right), \end{aligned} \tag{5.6}$$

where

$$p\left(x_{n,1}\right) = \frac{\tau_1 \exp\left\{-\frac{\tau_1}{2}\left(x_{n,1}-\mu_1\right)^2\right\}}{\tau_1 \exp\left\{-\frac{\tau_1}{2}\left(x_{n,1}-\mu_1\right)^2\right\} + \tau_2 \exp\left\{-\frac{\tau_2}{2}\left(x_{n,1}-\mu_2\right)^2\right\}},$$

and the damped cosines are parametrised by $\beta_1 = (-0.2, 0.6)'$, $\beta_2 = (-0.2, 0.4)'$ with $\sigma = 0.05$, and our mixture model is parametrised by $\tau_1 = \tau_2 = 0.8$, $\mu_1 =$

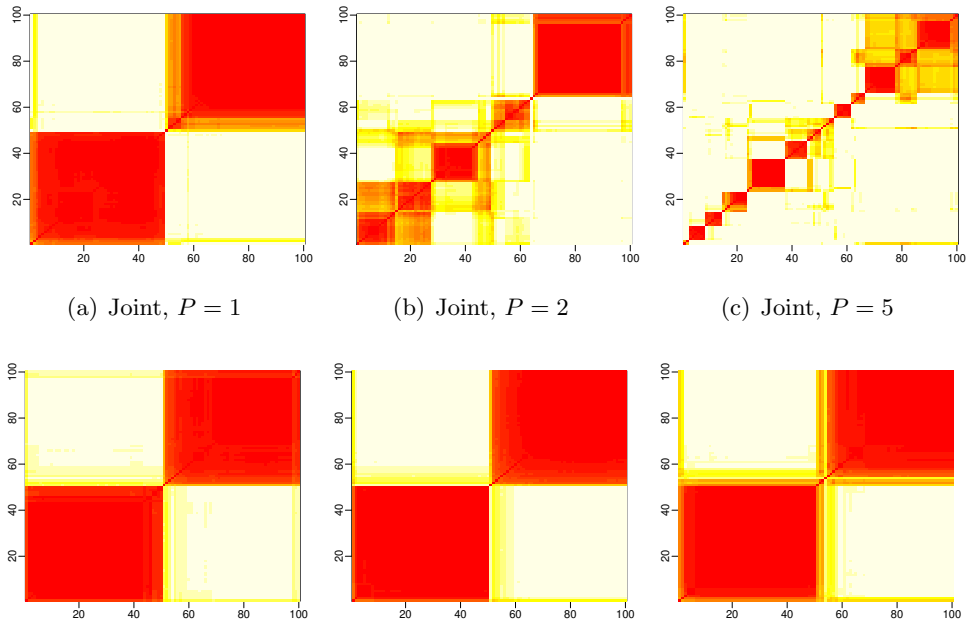
3, and $\mu_2 = 5$. The covariates are independently sampled from a multivariate normal $x_n \sim \mathrm{N}(\mu, \Sigma)$, centred at $\mu = (4, \ldots, 4)$, with standard deviation of 2 along each dimension, that is $\Sigma_{h,h} = 4$. The covariance matrix Σ models two groups of covariates: those in the first group are positively correlated among each other and the first covariate, but independent of the second group of covariates, which are positively correlated among each other but independent of the first covariate. In particular, we take $\Sigma_{h,l} = 3.5$ for $h \neq l$ in $\{1, 2, 4, \ldots, 2\lfloor p/2 \rfloor\}$ or $h \neq l$ in $\{3, 5, \ldots, 2\lfloor (p-1)/2 \rfloor + 1\}$, and $\Sigma_{h,l} = 0$ for all other pairs of $h \neq l$. The true data generating density function for y given the first input, in (5.6), is shown in Fig 5.5(a).

We consider Gaussian experts with identity link and ARD (as introduced in section 2.3.1) squared exponential kernels for the GPs with a $\mathrm{Ga}(1, 1)$ prior on the first input dimension length-scale, $\mathrm{Ga}(10, 1)$ prior on the other input dimension length-scales and a $\mathrm{Ga}(2, 1.5)$ prior on the magnitude. The constant means β_0 of the GPs have a $\mathrm{N}(0, 0.5^2)$ prior. The variance σ_y^2 has a log-$\mathrm{N}(\log(0.002), 0.2^2)$. For the joint model, the mass parameter has hyperparameters $(u_a = 1, v_a = 1)$, and for the enriched model, the mass parameters have hyperparameters $(u_\theta = 1, v_\theta = 1)$ and $(u_\psi = 1, v_\psi = 1)$. A Gaussian input model is used with hyperparameters of the conjugate normal-inverse gamma set to $u_0 = 4$, $c = 1/4$, $b_x = 1$, and $a_x = 2$. We perform posterior inference for both models by running the updating procedure outlined in the previous sections for 5000 total iterations with a burn-in period of 1000. Each chain was initialised with singleton clusters, including covariate sub-clusters for the enriched model.

A heat map of the posterior similarity matrix from the joint model, given in Fig. 5.1, highlights the two clusters for $P = 1$, with a greater number of clusters needed as P increases. Indeed, the VI estimate of the clustering (Fig. 5.2) contains two clusters for $P = 1$ and 10 clusters for $P = 5$. Conversely, we see that the enriched model highlights two y-clusters for each choice of P. Interestingly, the outlier with $x_1 = -1.93$ is allocated to the right cluster (red cluster in Figs. 5.2(d)-5.2(f)) in the VI estimate, although its allocation is uncertain.

For the enriched model, the x-level clustering requires an increasing number of clusters as P increases. Fig. 5.3 depicts the heat map of the posterior similarity matrix for the x-clustering with the two estimated y-clusters, and Table 5.1 reports the number of x-clusters in the VI estimated x-clustering within the two estimated y-clusters.

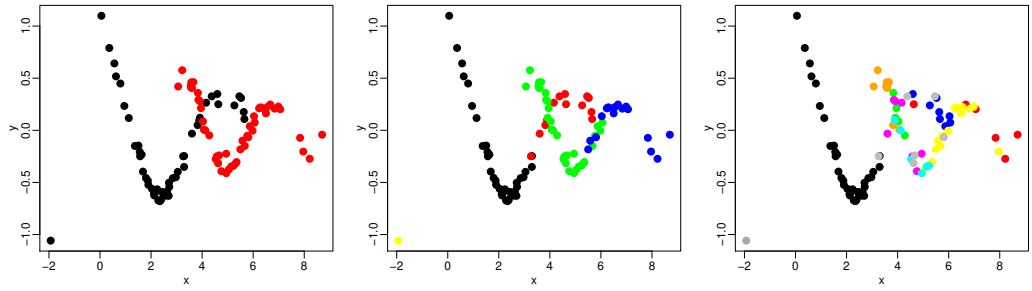
Conditioned on the VI clustering estimate, Fig. 5.4 depicts the allocation probabilities for a new test point $x_{*,1}$, with the other inputs marginalised, which



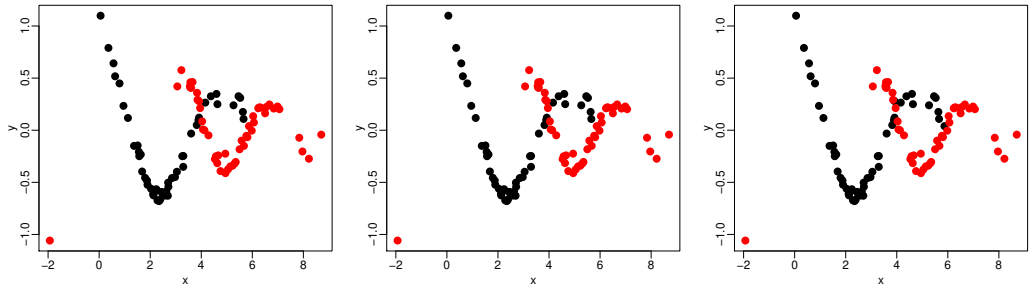
(a) Joint, $P = 1$ (b) Joint, $P = 2$ (c) Joint, $P = 5$

(d) Enriched, $P = 1$ (e) Enriched, $P = 2$ (f) Enriched, $P = 5$

Figure 5.1: Heat map of the posterior similarity matrix. The probability density represents the frequency with which two data points were indexed by same feature cluster. Each axis shows a re-ordered indexing set for the training samples. Rows corresponds to joint and enriched models respectively, whilst columns correspond to increasing $P = 1, 2, 5$.



(a) Joint, $P = 1$ (b) Joint, $P = 2$ (c) Joint, $P = 5$



(d) Enriched, $P = 1$ (e) Enriched, $P = 2$ (f) Enriched,$P = 5$

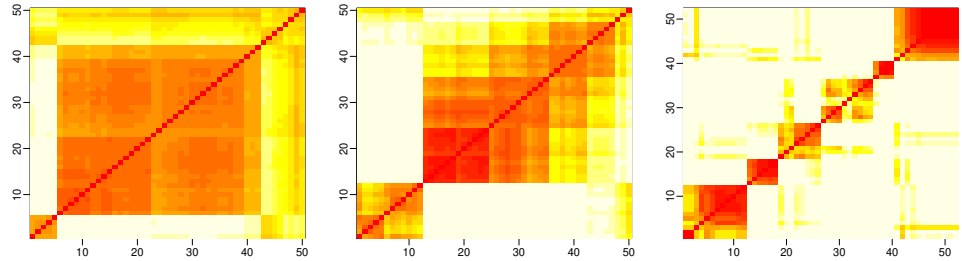
Figure 5.2: The VI clustering estimate with data points $(x_{n,1}, y_n)$ coloured by cluster membership. Rows corresponds to joint and enriched models respectively, whilst columns correspond to increasing $P = 1, 2, 5$.

y-cluster	$P = 1$	$P = 2$	$P = 5$
1	2	3	8
2	2	4	7

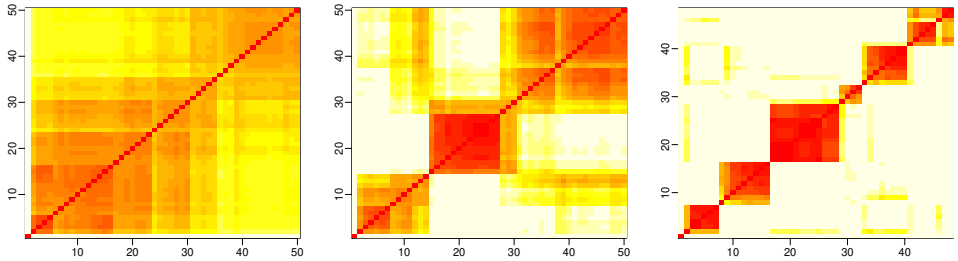
Table 5.1: The number of clusters in the VI estimated x-clustering within the two estimated y-clusters for the enriched model, as P increases.

represents the probability that the new point belongs to each of the estimated clusters or a new cluster as a function of the first input. For the enriched model, these allocation probabilities are conditioned on the two-level VI estimated clustering. As P increases, the joint model becomes increasingly uncertain of the allocation of new points with moderate values of $x_{*,1}$, while the enriched model is more robust to increasing P.

Summary statistics of the clustering are given in Table 5.2, including the VI distance between the true clustering $\boldsymbol{z}^t$ and VI estimated clustering $\widehat{\boldsymbol{z}}$, and the size of the 95% VI credible ball around the VI clustering estimate, denoted ϵ^*_{CB}. Notice that for the joint model with $P = 5$, the true clustering lies outside the 95% credible ball.



(a) $P = 1$: y-cluster 1 (b) $P = 2$: y-cluster 1 (c) $P = 5$: y-cluster 1

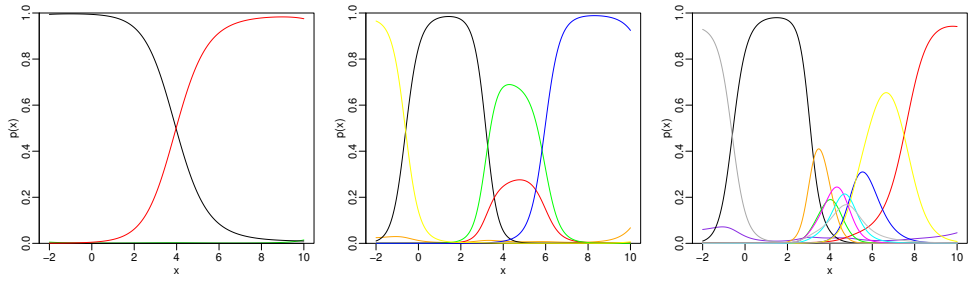


(d) $P = 1$: y-cluster 2 (e) $P = 2$: y-cluster 2 (f) $P = 5$: y-cluster 2

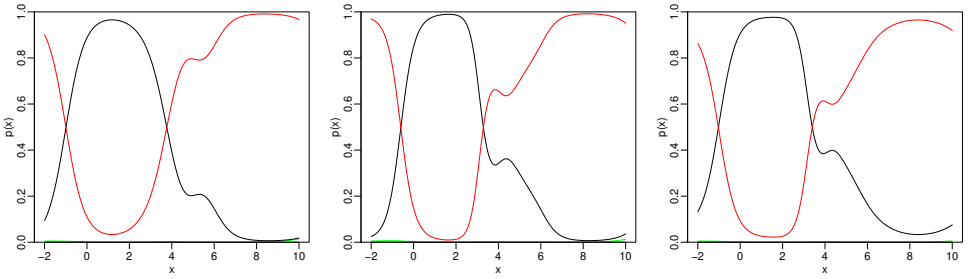
Figure 5.3: Heat map of the posterior similarity matrix for the x-clustering within the two estimated y-clusters for the enriched model. The probability density represents the frequency with which two data points were indexed by same covariate cluster. Each axis shows a re-ordered indexing set for the training samples. Rows correspond to y-cluster, whilst columns correspond to increasing $P = 1, 2, 5$.

We plot the estimates for the predictive response density and mean against the first covariate over a dense grid. These are presented in Fig. 5.5, for different choices of P. In the second and fourth rows the additional covariates are fixed to their sample means (approximately 4) for the joint and enriched models, respectively. Further, in the third and fifth rows, the additional covariates are marginalised. We compare the accuracy of each predictive response density with the approximate L_1 distance between estimated predictive response density and the true data generating density; this is then averaged across test samples, providing a Monte Carlo approximation for integration with respect to the true data generating distribution for x. These errors are given in Table 5.2. While errors generally worsen with increasing P, the enriched model is more robust.

Finally, coverage plots are presented in Fig. 5.6. Centred around the true values (sampled from the data generating distribution of (5.6)), these plots show the 95% highest posterior density credible interval for randomly sampled covariates (in some cases this may be a union of intervals). When the sample of the truth



(a) JmE: $P = 1$ (b) JmE: $P = 2$ (c) JmE: $P = 5$



(d) EmE: $P = 1$ (e) EmE: $P = 2$ (f) EmE: $P = 5$

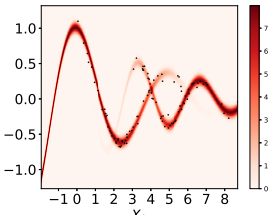
Figure 5.4: The allocation probabilities for a new test point $x_{*,1}$, with other covariates marginalised, conditioned on the estimated VI clustering, with colours corresponding to the estimated clusters. Rows corresponds to joint and enriched models respectively, whilst columns correspond to increasing $P = 1, 2, 5$.

lies within our credible interval the line is blue, otherwise it is red. The proportion of samples inside the interval is known as the coverage (denoted CI_{95}), which is reported in Table 5.2 alongside the average credible interval width (denoted $\bar{CI}_{95}$). We observe that the credible intervals for the joint model increase in width as P increases, whilst this behaviour is not observed in the enriched model.

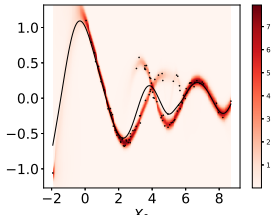
5.3.2 Alzheimer's Disease Neuroimaging Initiative challenge

Motivated by the Alzheimer's Disease Big Data DREAM Challenge competition[1], the aim of this study is to predict the change in cognitive scores 24 months after initial assessment. This will help predict the cognitive trajectory of patients, potentially assisting in early diagnosis of the Alzheimer's disease (AD). Earlier identification is particularly important in clinical trials designed to test the effectiveness of any proposed drugs or therapies, as treatments are expected to be most effective in early stages of the disease. Training data for the challenge was extracted from

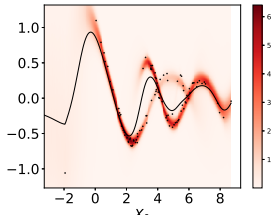
[1] `https://www.synapse.org/#!Synapse:syn2290704/wiki/60828`



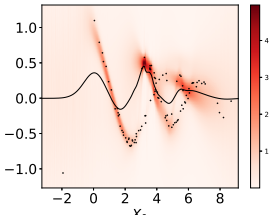
(a) True



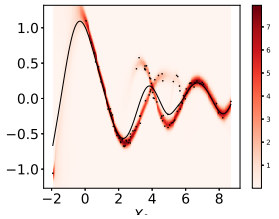
(b) Joint conditional, $P = 1$.



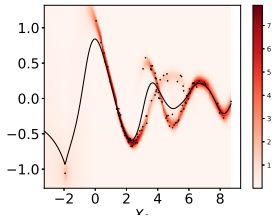
(c) Joint conditional, $P = 2$.



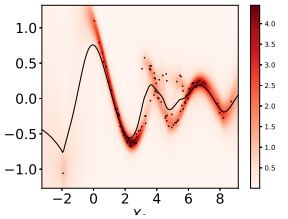
(d) Joint conditional, $P = 5$.



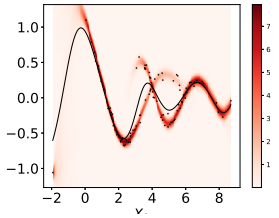
(e) Joint marginal, $P = 1$.



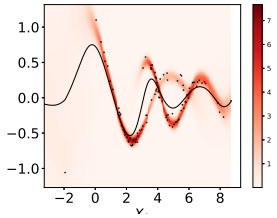
(f) Joint marginal, $P = 2$.



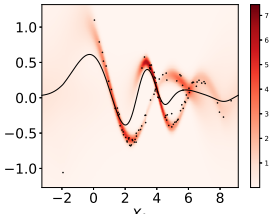
(g) Joint marginal, $P = 5$.



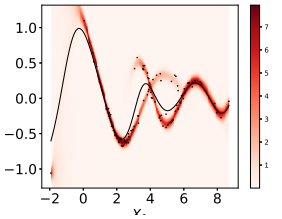
(h) Enriched conditional, $P = 1$.



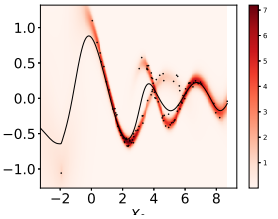
(i) Enriched conditional, $P = 2$.



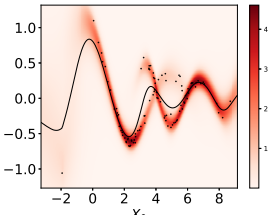
(j) Enriched conditional, $P = 5$.



(k) Enriched marginal, $P = 1$.

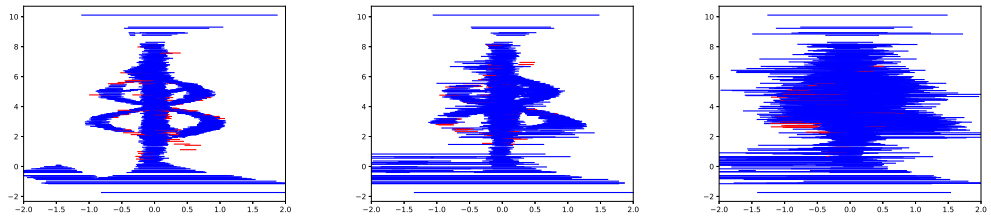


(l) Enriched marginal, $P = 2$.

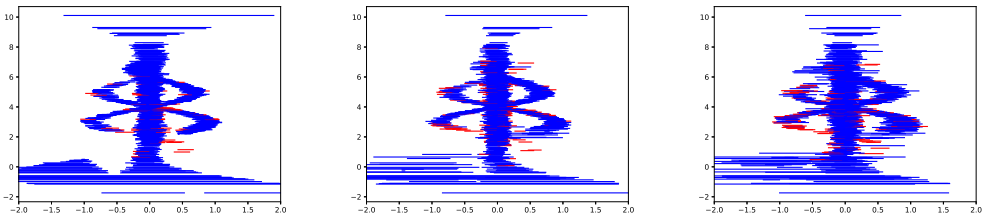


(m) Enriched marginal, $P = 5$.

Figure 5.5: Predictive density plots for the joint and enriched models for a grid of $x_{*,1}$ values, with additional covariates conditioned on their sample means (second and fourth rows) or marginalised (third and fifth rows), with increasing $P = 1, 2, 5$ (columns).



(a) JmE, $P = 1$ (b) JmE, $P = 2$ (c) JmE, $P = 5$



(d) EmE, $P = 1$ (e) EmE, $P = 2$ (f) EmE, $P = 5$

Figure 5.6: Coverage for the joint and enriched MoE with increasing $P = 1, 2, 5$. Each horizontal line depicts the 95% credible interval (based on quantiles with equal tails) and is blue if the sampled truth lies inside and red otherwise. The percentage of samples lying inside the interval is the coverage.

the Alzheimer's Disease Neuroimaging Initiative (ADNI) database[2]. The ADNI database contains neuroimaging, biological, and clinical data, along with summaries based on analyses of the neuroimages and biological data.

Our focus is on the first part of Question 1 of the challenge, that is, to predict the change in cognitive scores 24 months after initial assessment based on clinical data. The dataset consists of 767 participants with ordinal response y_n denoting

[2] The ADNI was launched in 2003 by the National Institute on Aging (NIA), the National Institute of Biomedical Imaging and Bioengineering (NIBIB), the Food and Drug Administration (FDA), private pharmaceutical companies and non-profit organizations, as a $ 60 million, 5-year public- private partnership. The primary goal of ADNI has been to test whether serial magnetic resonance imaging (MRI), positron emission tomography (PET), other biological markers, and clinical and neuropsychological assessment can be combined to measure the progression of mild cognitive impairment (MCI) and early Alzheimer's disease (AD). Determination of sensitive and specific markers of very early AD progression is intended to aid researchers and clinicians to develop new treatments and monitor their effectiveness, as well as lessen the time and cost of clinical trials. The Principal Investigator of this initiative is Michael W. Weiner, MD, VA Medical Center and University of California-San Francisco. ADNI is the result of efforts of many co-investigators from a broad range of academic institutions and private corporations, and subjects have been recruited from over 50 sites across the U.S. and Canada. The initial goal of ADNI was to recruit 800 adults, ages 55 to 90, to participate in the research, approximately 200 cognitively normal older individuals to be followed for 3 years, 400 people with MCI to be followed for 3 years and 200 people with early AD to be followed for 2 years. For up-to-date information, see `www.adni-info.org`.

		Clusters	VI($\boldsymbol{z}^t, \widehat{\boldsymbol{z}}$)	$\mathbb{E}$[VI($\boldsymbol{z}^t, \cdot$)$\vert\mathcal{D}$]	$\epsilon*_{\text{CB}}$	L_1	CI_{95}	$\bar{C}I_{95}$
	$P=1$	2	0.33	0.65	0.85	0.332	0.963	0.480
JmE	$P=2$	5	1.30	1.63	1.49	0.493	0.962	0.579
	$P=5$	10	2.67	3.07	1.72	0.939	0.966	1.11
	$P=1$	2	0.41	0.59	0.72	0.338	0.958	0.466
EmE	$P=2$	2	0.41	0.52	0.59	0.405	0.924	0.424
	$P=5$	2	0.65	0.67	0.86	0.643	0.907	0.478

Table 5.2: Columns (1)-(4) give summary statistics of the posterior over the clustering structure. (1) the number of clusters in the VI clustering estimate; (2) the VI distance between the true clustering $\boldsymbol{z}^t$ and the VI estimated clustering $\widehat{\boldsymbol{z}}$; (3) the posterior expected VI of the true clustering $\boldsymbol{z}^t$; and (4) the size of the 95% VI-credible ball (ϵ^*_{CB}). Column (5) gives the predictive accuracy as the approximate L_1 distance of the estimated and true conditional densities, whilst columns (6)-(7) give the coverage probabilities and average credible interval length.

mini-mental state exam (MMSE) scores at a 24 month follow-up visit. The MMSE is an extensively used clinical measure of cognitive decline and is defined on a 30 point scale, with lower scores reflecting increased impairment. The $P = 6$ inputs include one continuous input representing age (in fraction of years) at the initial assessment, one categorical discrete input denoting gender, and four ordinal discrete inputs. The four ordinal inputs consist of the MMSE scores at baseline; years of education; APOE genotype, recoded to reflect the number copies of the type 4 allele (i.e. takes values 0, 1, or 2); and diagnosis at baseline, taking values 0,1,2, or 3, to represent cognitively normal (CN), early mild cognitive impairment (EMCI), late mild cognitive impairment (LMCI), and Alzheimer's disease (AD), respectively.

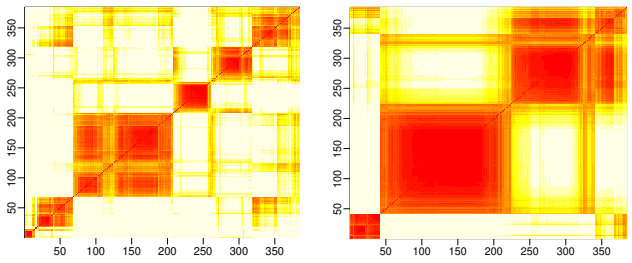
The enriched mixture of experts model allows us to flexibly recover non-linear trajectories of the cogntive decline of patients through Gaussian processes, while also clustering patients into covariate-dependent groups of similar trajectories. We consider the ordered probit link function with fixed cutoffs $0 = \varepsilon_0 < \varepsilon_1 = 1 < \varepsilon_2 = 2 \ldots < \varepsilon_{29} = 29$. The parametric local model for x_n, $p(x_n|\psi)$, is the product of one normal density for age, one categorical density for gender, and four binomial densities for baseline MMSE (with $G_3 = 30$), education (with $G_4 = 20$), APOE4 (with $G_5 = 2$), and diagnosis (with $G_6 = 3$).

We consider an ARD squared exponential kernel for GP with $\text{Ga}(a_{l,p}, b_{l,p})$ priors on the length-scales with $a_l = (3, 2, 3, 5, 3, 2)$ and $b_l = (3/20, 5, 1, 1, 5, 4)$ (parametrised with expectation a_l/b_l) and a $\text{Ga}(a_m, b_m)$) prior on the magnitude with $a_m = 2$ and $b_m = 1$. These parameters were selected to reflect our prior knowledge on the relationship between follow-up MMSE and the inputs and based

on the range of the inputs. The GP is assumed to have a prior constant mean with a $\mathrm{N}(20, 7.5^2)$ prior. The variance σ_y^2 has a $\mathrm{Ga}(a_y, b_y)$ prior with $a_y = 1.5$ and $b_y = 0.75$. The mass parameter has hyperparameters $u_a = 1$, $v_a = 1$. The input hyperparameters are $u_0 = 72$, $c = 2$, $b_x = 10$, and $a_x = 2$ for age; $\gamma_2 = (1, 1)$ for gender; $\gamma_3 = (5, 1)$ for MMSE; $\gamma_4 = (3, 2)$ for education; $\gamma_5 = (1, 3)$ for APOE4; $\gamma_6 = (1, 1)$ for diagnosis.

	Clusters	$\mathbb{E}[\mathrm{VI}(\widehat{\boldsymbol{z}}, \cdot)\|\mathcal{D}]$	ϵ*CB	MAE_{test}	CI_{95}	$\bar{CI}_{95}$	$\mathrm{med}(CI_{95})$
JmE	7	1.75	2.21	2.131	0.953	9.16	8
EmE	3	1.07	1.38	**2.104**	0.950	8.99	8
GuanLab	3	-	-	2.153	-	-	
GuanLab2	3	-	-	2.208	0.945	11.06	11.34
ADDT	-	-	-	2.158	0.867	8.29	8.08

Table 5.3: Columns (1)-(3) give summary statistics of the posterior over the clustering structure. (1) the number of clusters in the VI clustering estimate; (2) the posterior expected VI of the estimated clustering $\widehat{\boldsymbol{z}}$; and (3) the size of the 95% VI credible ball around $\widehat{\boldsymbol{z}}$. Column (4) gives mean absolute error on the held out test data, whilst columns (5)-(6) give summary statistics of the empirical coverage.

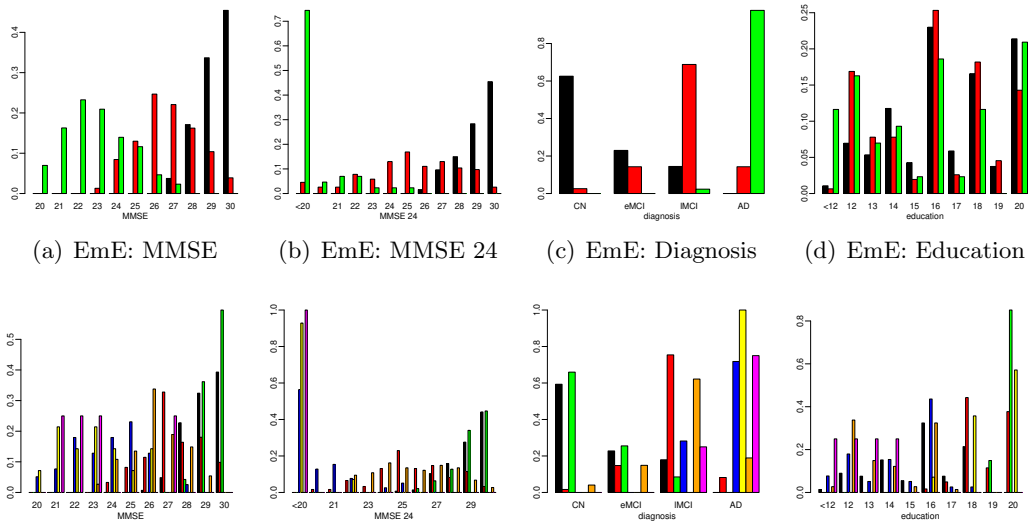


(a) Joint (b) Enriched

Figure 5.7: Heat map of the posterior similarity matrix for the y-clustering for the joint and enriched models. The probability density represents the frequency with which two data points were indexed by same covariate cluster. Each axis shows a re-ordered indexing set for the training samples.

Unfortunately, the test data used in the competition can no longer be accessed. As such, we have split the available data into training and test sets of sizes $N = 384$ and $N^* = 383$, respectively. The best performers[3] for this subchallenge were the GuanLab and ADDT teams. The model of team GuanLab (Zhu and Guan [2014]) separated training samples into three groups based on the diagnosis of CN, MCI or AD and trained support vector machines for each group. The ADDT model

[3] `https://www.synapse.org/#!Synapse:syn2290704/wiki/70719`



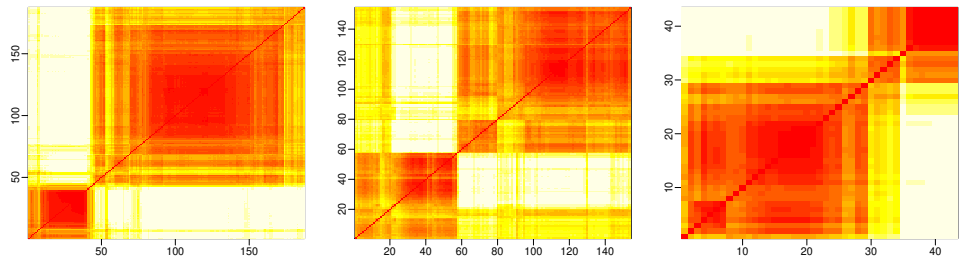
(a) EmE: MMSE (b) EmE: MMSE 24 (c) EmE: Diagnosis (d) EmE: Education

(e) JmE: MMSE (f) JmE: MMSE 24 (g) JmE: Diagnosis (h) JmE: Education

Figure 5.8: A visualization of the VI clustering estimate through side-by-side bar plots colored by cluster membership. Rows correspond to Joint and Enriched models respectively, whilst columns correspond to the MMSE baseline, MMSE 24-month follow-up, diagnosis and education.

(Hwang et al. [2014]) used robust regression based on M-estimation, replaced diagnosis and APOE4 with an optimal combination based on Spearman correlation, AIC and BIC, and included interaction terms. For comparison, we implemented the GuanLab model using the **svm** function of the **e1071** package in **R** (Meyer et al. [2018]) and the ADDT model using the **rlm** function of the **MASS** package in **R** (Venables and Ripley [2002]). Table 5.3 summarises the comparison of the model on the held out test data in terms of mean absolute error, the empirical coverage probability of the 95% confidence intervals for predictions and the average and median length of the 95% confidence intervals for predictions. As the **svm** function provides only predictions, in GuanLab2, we train a linear regression model within group to obtain confidence intervals for predictions.

We first note that the enriched model performs slightly better than the joint model in terms of mean absolute test error and maintains good coverage with smaller uncertainty, reflected in a reduction in the length of the HPD credible intervals (see Table 5.3). Posterior medians, i.e. the point estimate under the absolute error loss, are used to predict MMSE scores, which are appropriate due to the heavy left tail of the predictive densities. The improvement of the enriched model is due to the ability to capture the relationship between y and x with fewer clusters, also leading to a more interpretable clustering structure. Indeed, the VI clustering estimate for



(a) y-cluster 1 (b) y-cluster 2 (c) y-cluster 3

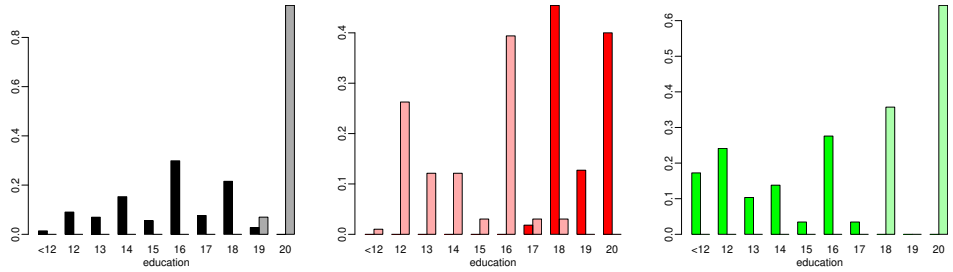
Figure 5.9: Heat map of the posterior similarity matrix for the x-clustering within the three estimated y-clusters for the enriched model. The probability density represents the frequency with which two data points were indexed by same covariate cluster. Each axis shows a re-ordered indexing set for the training samples.

the enriched model has only three clusters, while the VI clustering estimate for the joint model has seven clusters.

The clusterings for the two models are summarised in Table 5.3 with heat maps of the posterior similarity matrix in Fig. 5.7 and visualization of the VI clusterings through side-by-side bar plots of MMSE baseline, MMSE follow-up, diagnosis, and education with colours representing clusters. Interestingly, the enriched model identifies three clusters consisting mostly of cognitively normal (black), mild cognitive impairment (red), and AD (green) individuals, similar to the GuanLab model, with slight modifications considering the other variables, particularly, MMSE baseline and follow-up scores. For example, one late MCI individual is allocated to the AD (green) cluster in Fig. 5.8(c) due to the observed sharp drop in MMSE from 27 at baseline to 8 at follow-up. Additional VI cluster visualisations based on APOE4, gender, and age (not shown) show that the relative proportion of individuals in the red and green clusters increases slightly with higher APOE4, but does not (marginally) depend on gender and age.

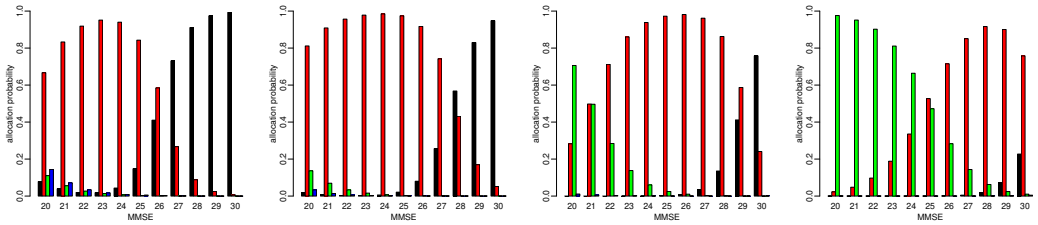
The joint model, on the other hand, further subdivides clusters due to multimodality in education. Similarly, for the enriched model, the VI estimate of x-clustering within each VI estimated y-cluster, contains two x-clusters due to multimodality in education. Fig. 5.9 depicts the heap map of the posterior similarity matrix for the x-clustering within each estimated y-cluster and Fig. 5.10 shows the VI estimate of x-clustering within each VI estimated y-cluster for education, with each estimated x-clustering containing two clusters.

We can further appreciate the difference between the deterministic clustering of the GuanLab model and the stochastic clustering of the enriched model in Fig.



(a) y-cluster 1: Education (b) y-cluster 2: Education (c) y-cluster 3: Education

Figure 5.10: A visualization of the VI x-clustering estimate with each y-cluster through side-by-side bar plots for education. Colour corresponds to the y-cluster, while shading corresponds to the x-cluster.

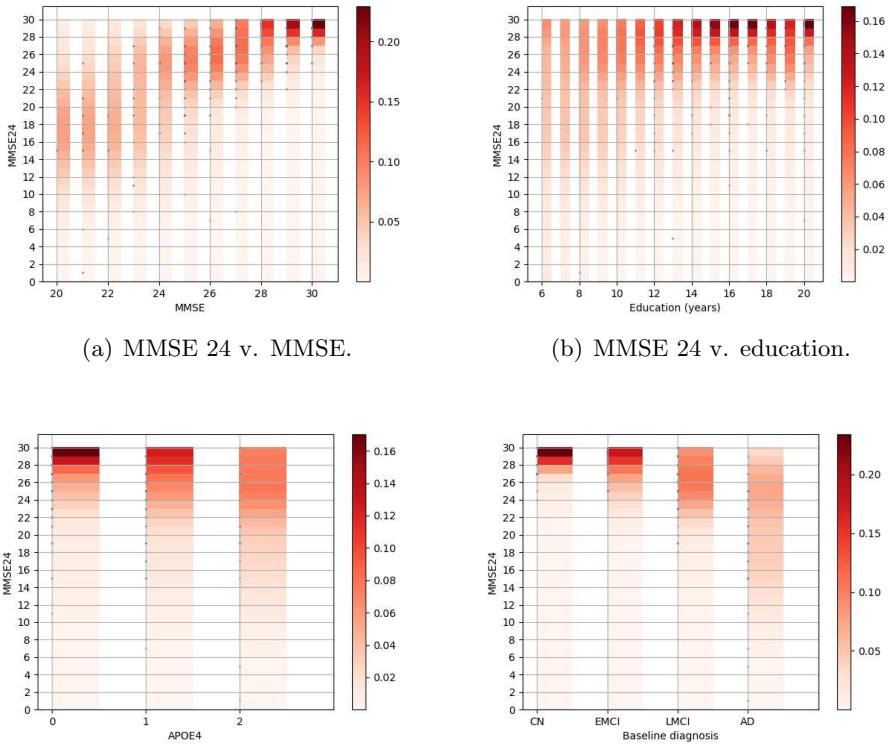


(a) Cogntively normal (b) early MCI (c) late MCI (d) Alzheimer's disease

Figure 5.11: The allocation probabilities for a new test point as a function of baseline MMSE and diagnosis of CN in (a), eMCI in (b)., lMCI in (c), and AD in (d), with other covariates marginalised. Allocation probabilities are based on the estimated VI clustering and coloured by cluster membership for each of the estimated VI clusters from the enriched model in Fig. 5.8.

5.11, which shows the allocation probabilities of a new test point for MMSE baseline scores of 20-30 and diagnosis of CN (Fig. 5.11(a)), eMCI (Fig. 5.11(b)), lMCI (Fig. 5.11(c)), AD (Fig. 5.11(d)), with other covariates marginalised. As opposed to the GuanLab model which classifies new individuals based on diagnosis, we observe that CN individuals with baseline MMSE ≥ 27 have the highest probability of being allocated to the black cluster, while this baseline MMSE cutoff is increased to 28 and 30 for eMCI and lMCI individuals, respectively. Below these respective cutoffs, CN, eMCI, and lMCI individuals have the highest probability of being allocated to the red cluster (apart from lMCI individuals with baseline MMSE of 20 that are allocated to the green cluster with highest probability). Instead, AD individuals have the highest probability of belonging to the red cluster for baseline MMSE ≥ 25 and to the green cluster otherwise. We note that for CN individuals with low MMSE baseline (not observed), there is a small probability of allocation to a new (blue)

cluster.



(a) MMSE 24 v. MMSE.

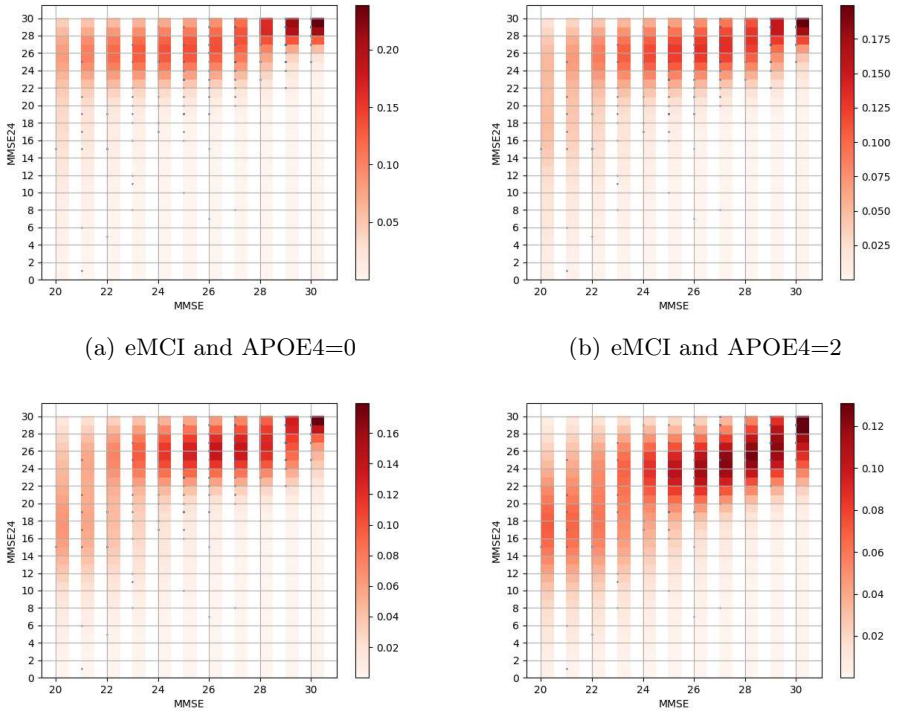
(b) MMSE 24 v. education.

(c) MMSE 24 v. APOE4.

(d) MMSE 24 v. diagnosis.

Figure 5.12: Marginalised predictive distribution for MMSE 24-month follow-up as a function of MMSE baseline, education, APOE4, and the baseline diagnosis for the enriched mixture of experts model.

For the enriched model, the predictive densities within each cluster are averaged with respect to these allocation probabilities and the posterior on the clustering to produce densities and credible intervals that change smoothly with the inputs. Specifically, Fig. 5.12 shows how the marginal predictive densities become less peaked and the credible intervals increase for decreased baseline MMSE, decreased education, increased APOE4, and increased severity in diagnosis; and Fig. 5.13 shows how the predictive densities as a function baseline MMSE interact with diagnosis and APOE4, with a greater decrease in follow-up scores and more uncertainty for more severe dementia type and increased APOE4. The GuanLab and ADDT models, on the other hand, are not able to change smoothly, with, for example, a minimum prediction interval length of 8 for ADDT, despite the high concentration of follow-up MMSE scores close to 30 for CN individuals with a baseline MMSE of



(a) eMCI and APOE4=0

(b) eMCI and APOE4=2

(c) lMCI and APOE4=0

(d) lMCI and APOE4=2

Figure 5.13: Predictive distribution for MMSE 24-month follow-up as a function of MMSE baseline, diagnosis, and APOE4, with other covariates marginalised for the enriched model. Columns represent APOE4 types of 0 and 2, whilst rows represent diagnosis of early and late MCI.

30. Thus, the enriched model, in addition to slightly improved mean absolute test error, provides much improved uncertainty quantification in predictions, which is particularly important in clinical settings and in relation to established cutoffs for MMSE (scores of 20-24 suggest mild dementia, 13-20 suggest moderate dementia, ≤ 12 indicate severe dementia).

5.4 Discussion

Infinite mixtures of GP experts are highly flexible Bayesian nonparametric models, that can be used to model non-stationary functions and capture departures from the typical homeoscedastic normality assumptions on the errors. In this work, we proposed a novel enriched mixture of generalised Gaussian process experts that makes local independence assumptions on the inputs, to increase scalability and

allow inclusion of multiple input types, and utilises a nested partitioning scheme, to increase predictive accuracy, improve uncertainty quantification and provide a more interpretable clustering structure. Moreover, through the generalised Gaussian process framework, we can account for different output types.

We have developed efficient MCMC algorithms for posterior inference based on the analytically computable allocation rule of the enriched Dirichlet process. However, these MCMC algorithms can be quite slow, particularly for increasing sample size and non-continuous outputs. In fact, computation times ranged from hours, for the simulated examples with small sample sizes and continuous outputs, to days, for the larger real data examples with non-continuous outputs (on a single computer). An important future direction will focus on fast approximate inference techniques, specifically on the MAP inference techniques developed in Raykov et al. [2014] that maintain a non-degenerate likelihood, enabling out-of-sample predictions and the use of standard tools such as cross-validation. Moreover, the MAP inference scheme can be easily derived from the Gibbs sampling scheme, for closed-form marginal likelihoods. For large sample sizes, further computational gains can be made through sparse or low-rank assumptions on the GP experts, (see e.g. Rasmussen and Williams [2005], chapter 8).

In order to scale to higher dimensional input spaces without degrading the predictive performance, the local independence and nested partitioning assumptions are crucial. However, to scale the GP experts to higher dimensional input spaces further assumptions are needed. In future research, we plan to investigate dimensionality reduction for GPs (Snelson and Ghahramani [2006b]) and the incorporation of variable selection techniques for DP mixtures (Papathomas et al. [2012] and Barcella et al. [2017]), as more flexible approaches to scale the GP experts to higher dimensional input spaces.

Chapter 6

Conclusion

This thesis considered a variety of Bayesian nonparametric models for predictive modelling. Firstly, a fully Bayesian inference scheme for the Gaussian process latent variable model was presented. This is a highly flexible model, with uncertainty manifesting in many components. This motivated a framework for quantification and propagation of this uncertainty through the model.

An existing procedure, based on variational inference, gives a principled approach towards this end. However, the scheme is unable to capture hyperparameter uncertainty, and makes strong distributional assumptions over the latent variables. This procedure approximates hyperparameters by optimising a lower bound to the marginal likelihood. This has been shown to lead to poor quantification of uncertainty; where the smallest uncertainty may appear where the approximation is poorest, and hyperparameter bias increases with the number of hyperparameters.

To address this a Markov chain Monte Carlo scheme was developed that allowed for accurate quantification of these uncertainties, with asymptotic guarantees of convergence. This was made challenging by the strong correlations that exist between latent variables and hyperparameters, which was overcome using an unbiased pseudo estimate for the marginal likelihood that approximately integrated over the latent variables in a collapsed Gibbs sampler. This sampler was then uncollapsed using elliptical slice sampling. The scheme was demonstrated on a simulated example, demonstrating the improved accuracy and uncertainty quantification in predictions when compared with the variational approach. Further cases of this example shed light on situations where the variational approximation may work well, and where it is inadequate.

Following this, an emulator for groundwater flow models was proposed to address the forward problem of uncertainty quantification. It is often the case that

dynamics of a fluid are modelled over a spatial domain, with each sample (corresponding to a unique input/parametrisation) containing values defined over a dense spatial mesh. Given the challenges of performing inference in a high dimensional space, this motivated a framework that incorporates manifold learning with emulation, allowing inference to be performed on latent projections of model outputs.

The manifold learning was performed using LTSA, a nonparametric method that provides an automatic pre-image map. Analytical forms for the output distribution were obtained by pushing the feature-space Gaussian distribution through a locally linear reconstruction map. Additionally, analytical estimates of the moments of the predictive distribution were derived by approximate marginalisation of the stochastic input.

This framework was applied to two models for groundwater contamination using the Karhunen-Loève expansion for a log-normally distributed stochastic hydraulic conductivity input field. The first example considered a linear, steady state Darcy's Law with a contaminant mass balance in a 2-d domain, and the second example considered a time-dependent Richards equation evaluated at fixed time in a 3-d domain. The ability to accurately predict outputs as well as the moments was then demonstrated.

Finally, an enriched clustering model for generalised mixtures of Gaussian process experts was presented, which partitions the input space into regions where stationary and error assumptions of the GP must only hold locally. Local independence assumptions of the inputs were made, which allowed for the inclusion of multiple input types, and better scaling with increasing dimensions. Additionally, the enriched Dirichlet process was utilised, allowing for a nested partitioning scheme which prevented the creation of an unnecessary number of experts, and allowed for an analytically computable allocation rule, which enabled the development of efficient sampling algorithms for posterior inference for probabilistic modelling of uncertainty. These advantages were demonstrated on a highly non-linear toy example with increasing input dimensions, and an Alzheimer's Disease Neuroimaging Initiative challenge with the aim of improving prediction of decline in cognitive impairment.

Appendix A

Supplementary material for Chapter 3

In this section additional information relevant to the material of Chapter 3 is presented.

A.1 Simulated sinusoidal example

First shown are figures which demonstrate the mixing of the collapsed Gibbs sampler, including trace plots (see Fig. A.1) and autocorrelation plots (see Fig. A.2) for each hyperparameter and across each chain.

The conditional latent posterior distribution is then shown given the hyperparameters at state 820, $\theta^{(820)}$, of the collapsed Gibbs sampler (see Fig. A.3) and given the set of maximum marginal likelihood hyperparameters, $\theta^{(ML)}$, obtained from jointly optimising over latent variables and hyperparameters (see Fig. A.4). These figures compare the quality of the variational approximation used in VEM to the true posterior used for predictions with the proposed PM inference scheme. Due to the high dimensional nature of these spaces, only bivariate contours, corresponding to two training samples, can be visualised at a time. In figures A.5 and A.6 the marginal conditional latent distribution for each sample is plotted alongside each other, given $\theta^{(820)}$ and $\theta^{(ML)}$ respectively. These figures demonstrate a clear tendency for the variational approximation to underestimate the variance and approximate local modes. Also shown in figures A.7 to A.10 are the marginal distributions for the benchmark examples, given the approximate maximum marginal likelihood hyperparameters.

Following this the predictive distributions using both the VEM and PM

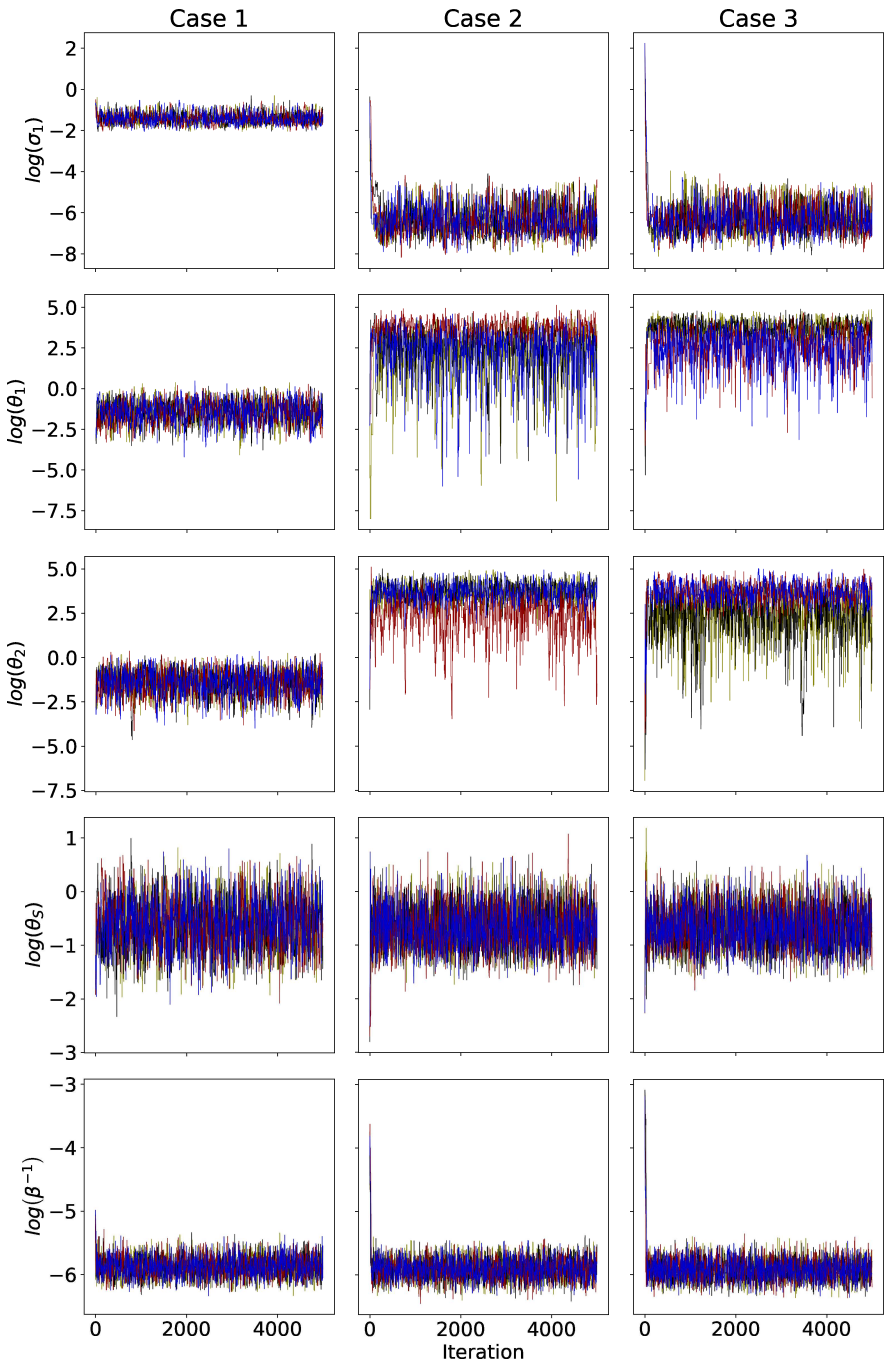


Figure A.1: Trace plots for the collapsed Gibbs hyperparameter posterior samples (with no thinning applied). The three columns correspond to the three data generating cases, while each row corresponds to a different hyperparameter.

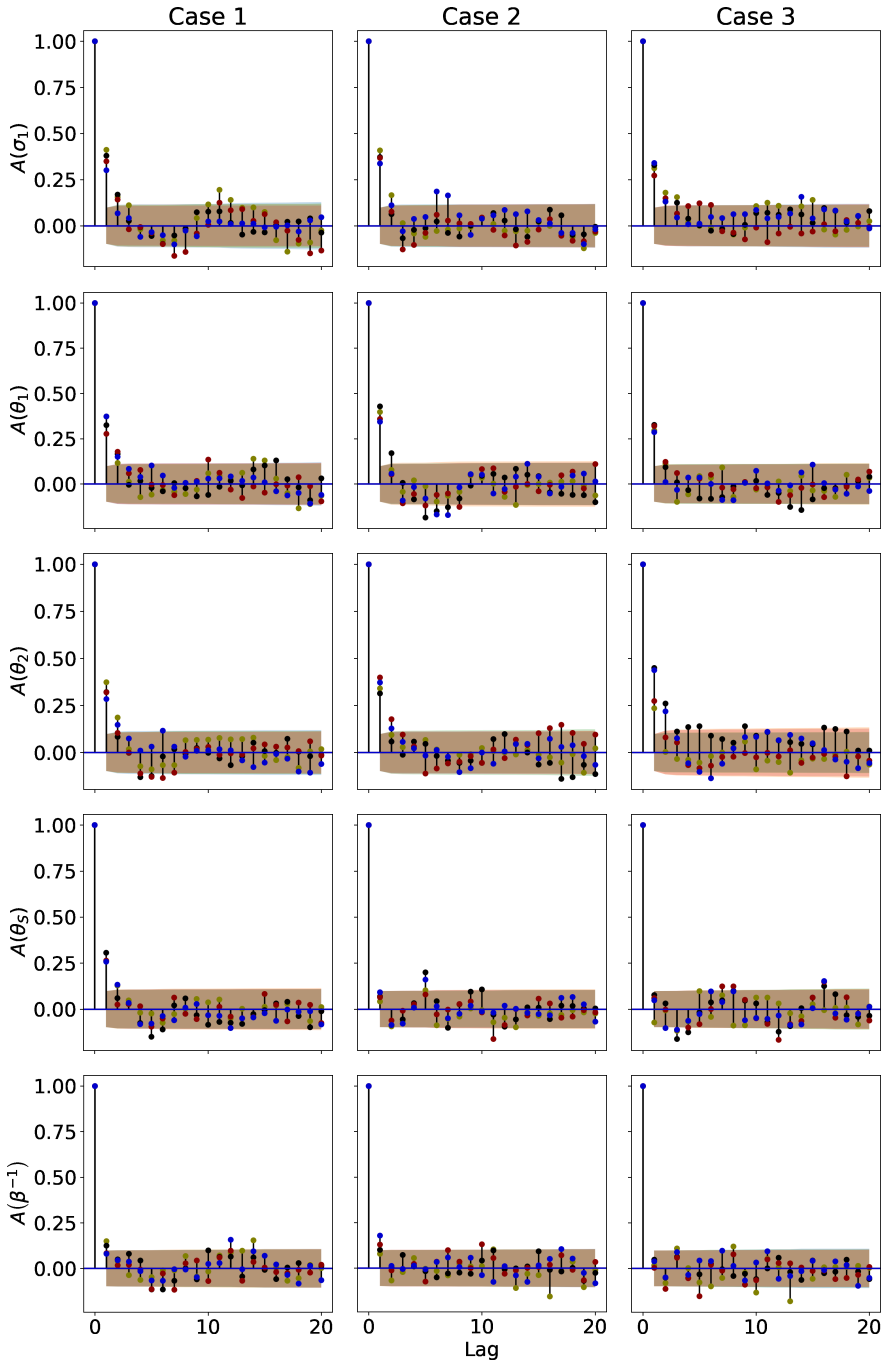


Figure A.2: Auto-correlation lag plots of the collapsed Gibbs hyperparameter posterior samples after a thinning factor 10 is applied. The three columns correspond to the three data generating cases, while each row corresponds to a different hyperparameter.

approach are compared to the true data generating distribution for each case in figures A.11 to A.13. As shown in the main manuscript, the PM inference scheme more accurately captures the true data generating function. The VEM inference scheme underfits for each example, with this behaviour becoming more extreme as the cases become increasingly misspecified.

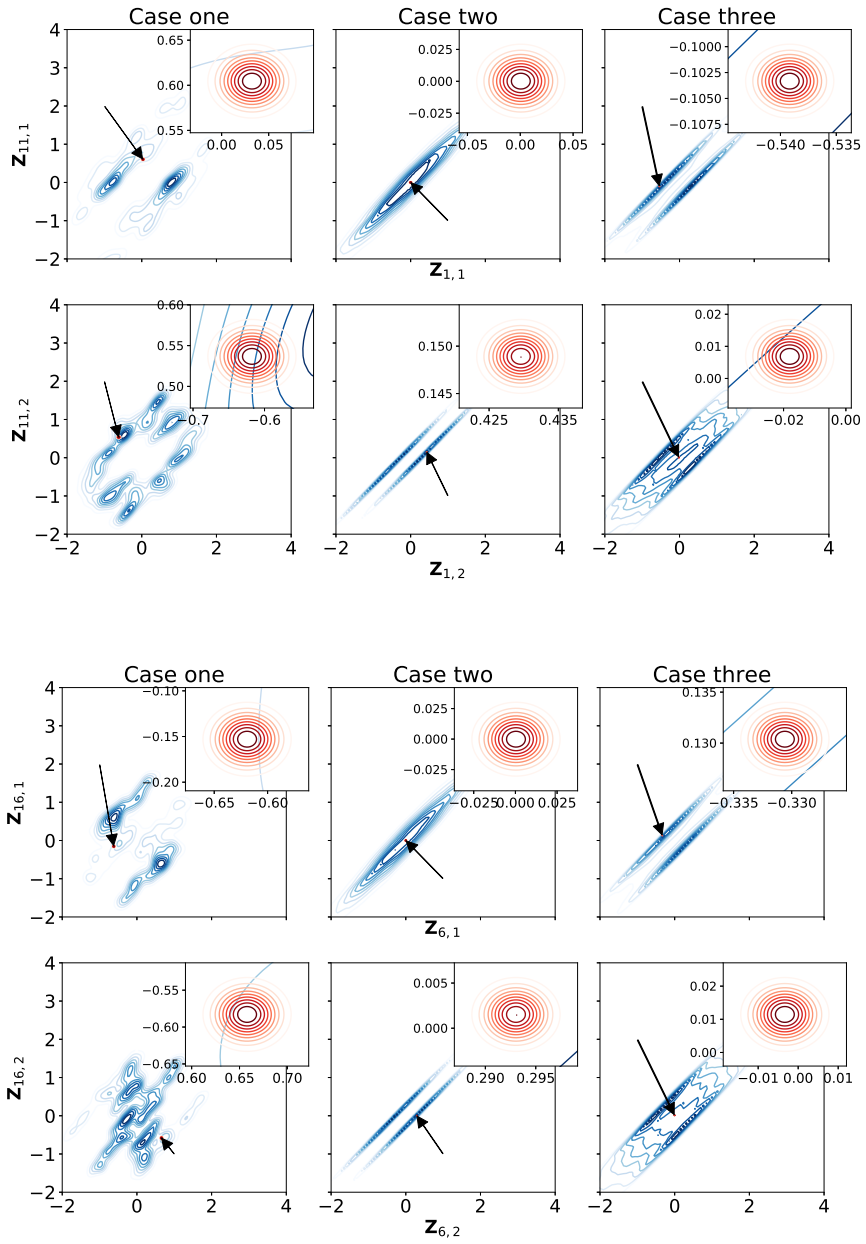


Figure A.3: PM inference scheme. Bivariate marginal latent posterior distributions for sample pairs $(1, 11)$ (top two rows) and $(6, 16)$ (bottom two rows), conditional on hyperparameter posterior sample $\theta^{(820)}$. The exact posterior (in blue) is obtained using kernel density estimation on $100,000$ elliptical slice samples, and the variational approximation (in red) is known analytically. The three columns correspond to the three data generating cases. The first and third rows correspond to the first latent dimension, while the second and fourth rows correspond to the second latent dimension.

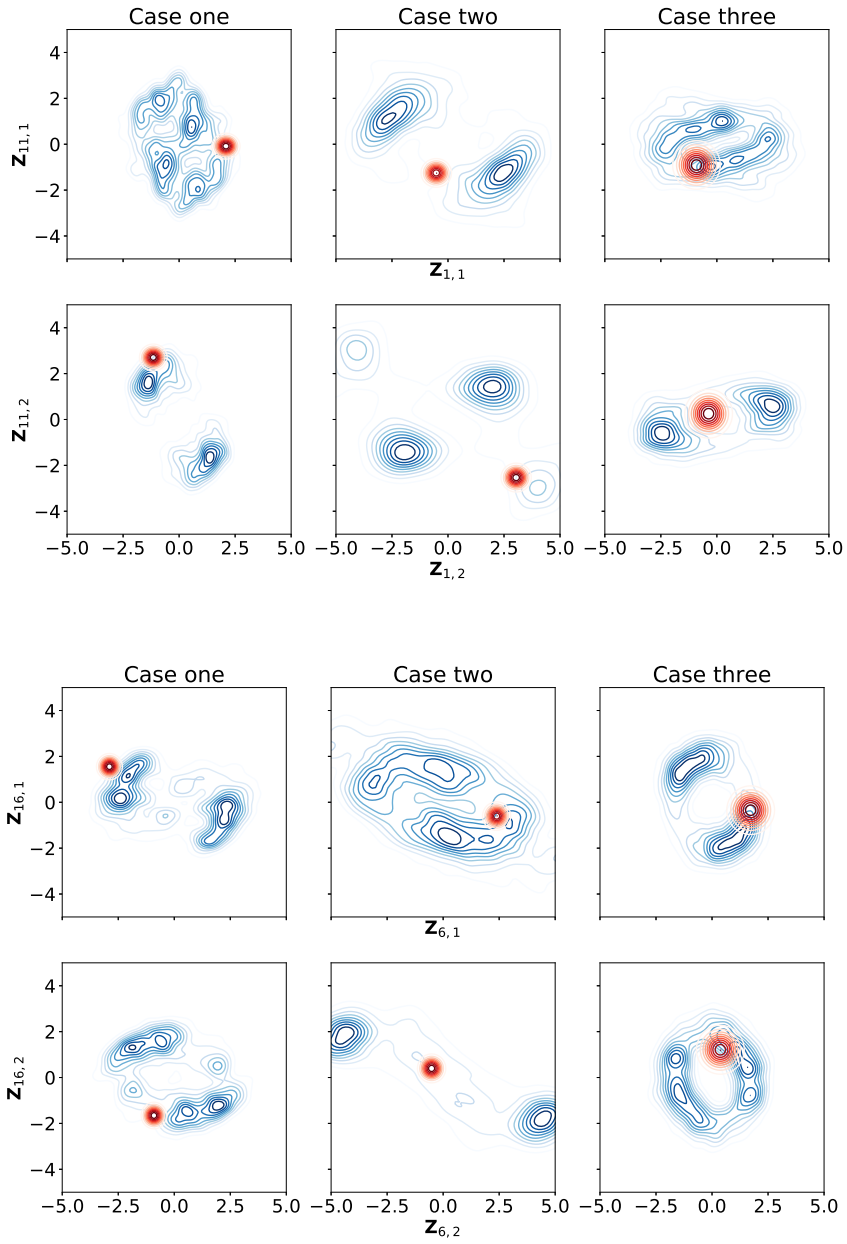


Figure A.4: VEM inference scheme. Bivariate marginal latent posterior distributions for sample pairs $(1, 11)$ (top two rows) and $(6, 16)$ (bottom two rows), conditional on hyperparameter posterior sample $\theta^{(ML)}$. The exact posterior (in blue) is obtained using kernel density estimation on $100,000$ elliptical slice samples, and the variational approximation (in red) is known analytically. The three columns correspond to the three data generating cases. The first and third rows correspond to the first latent dimension, while the second and fourth rows correspond to the second latent dimension.

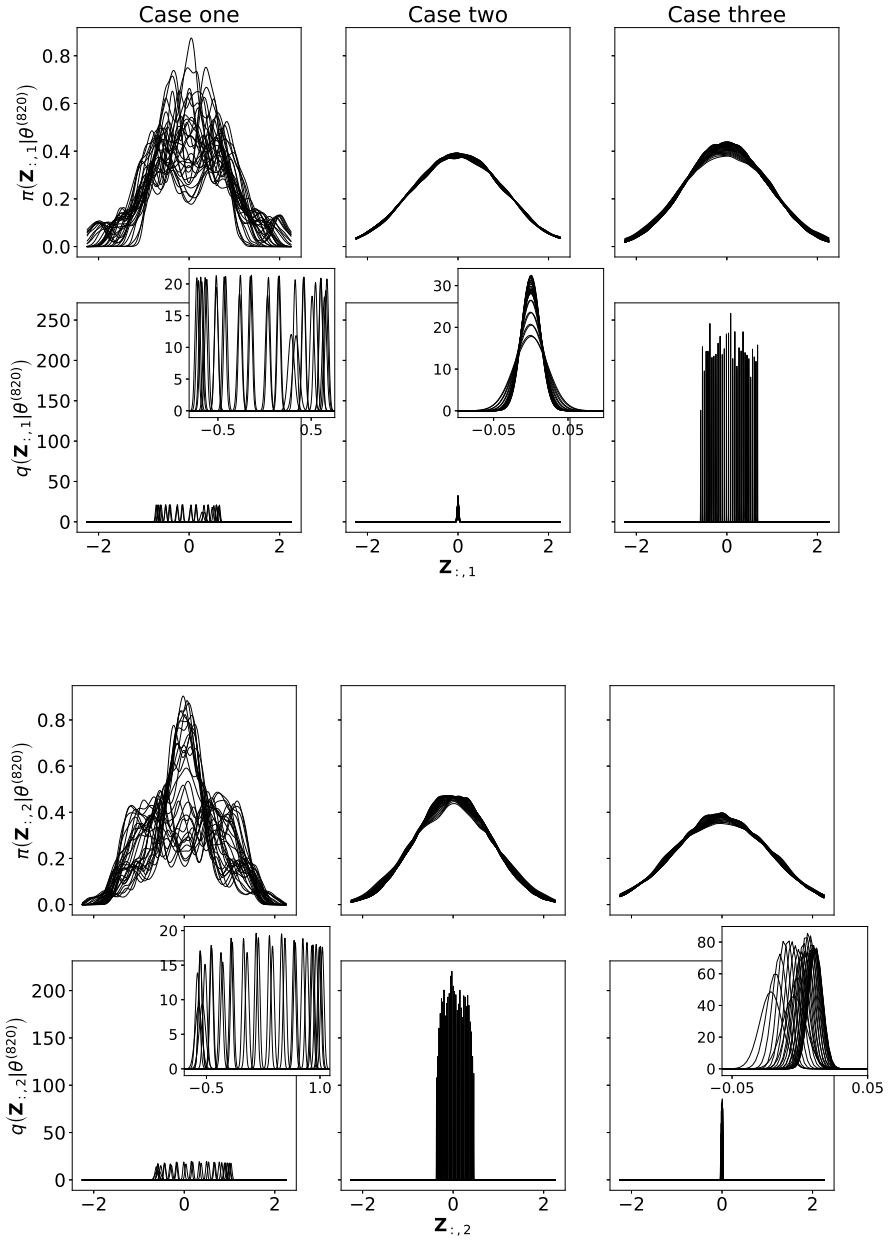


Figure A.5: PM inference scheme. Marginal latent posterior distributions for all samples, conditional on hyperparameter posterior sample $\theta^{(820)}$. The exact posterior (first and third row) is obtained using Kernel Density Estimation on the ESS samples, and the variational approximation (second and fourth row) is known analytically. The three columns correspond to the three data generating cases. The first two rows corresponds to the first latent dimension, whilst the last two rows refer to the second latent dimension.

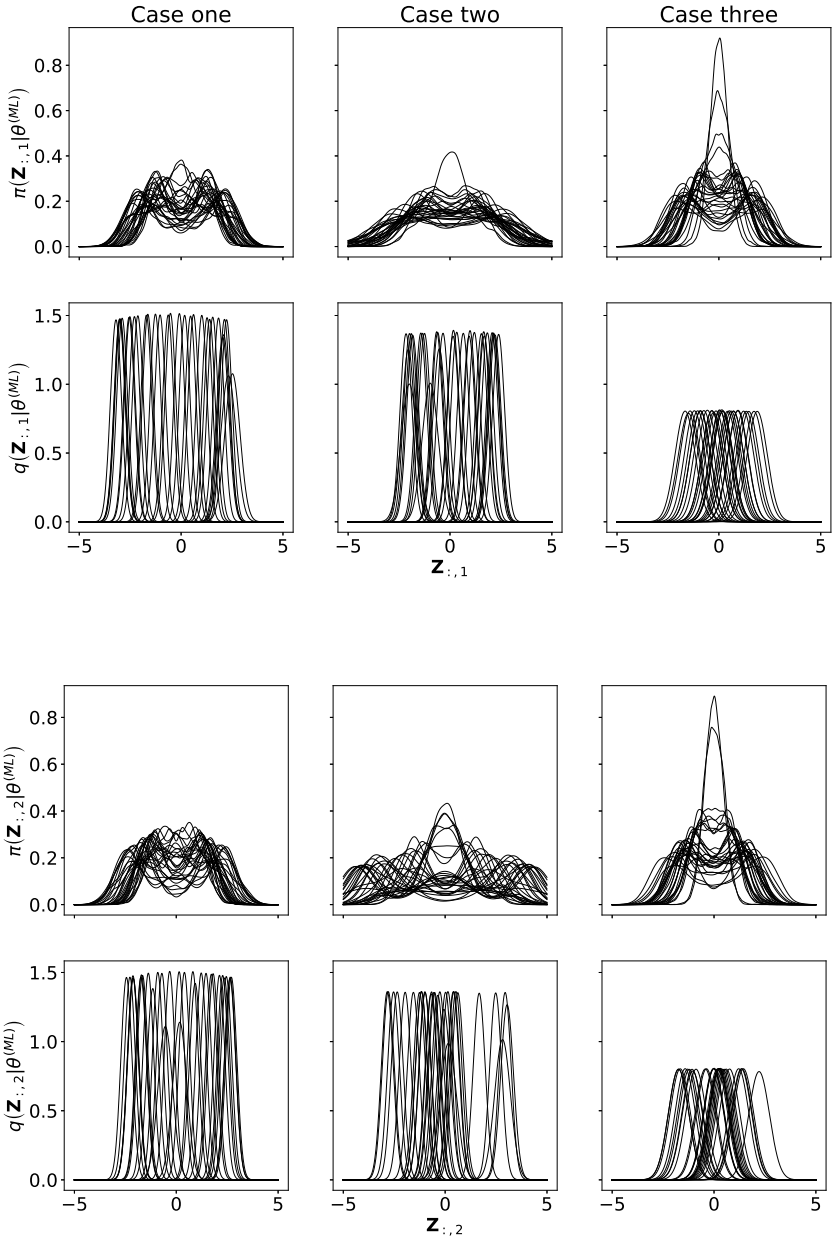


Figure A.6: VEM inference scheme. Marginal latent posterior distributions for all samples, conditional on the approximate maximum marginal likelihood hyperparameters $\theta^{(ML)}$. The exact posterior (first and third row) is obtained using Kernel Density Estimation on the ESS samples, and the variational approximation (second and fourth row) is known analytically. The three columns correspond to the three data generating cases. The first two rows corresponds to the first latent dimension, whilst the last two rows refer to the second latent dimension.

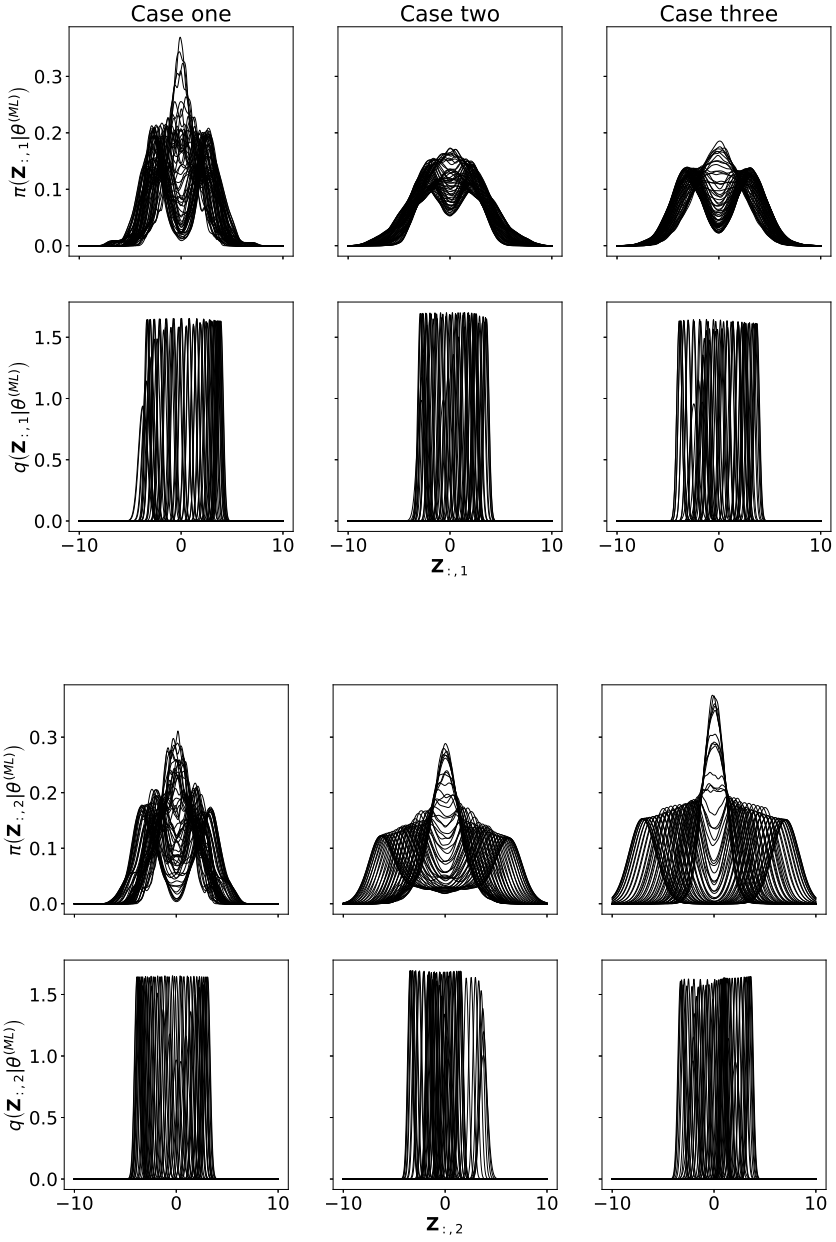


Figure A.7: VEM inference scheme with $N = 60$. Marginal latent posterior distributions for all samples, conditional on the approximate maximum marginal likelihood hyperparameters $\theta^{(ML)}$. The exact posterior (first and third row) is obtained using Kernel Density Estimation on the ESS samples, and the variational approximation (second and fourth row) is known analytically. The three columns correspond to the three data generating cases. The first two rows corresponds to the first latent dimension, whilst the last two rows refer to the second latent dimension.

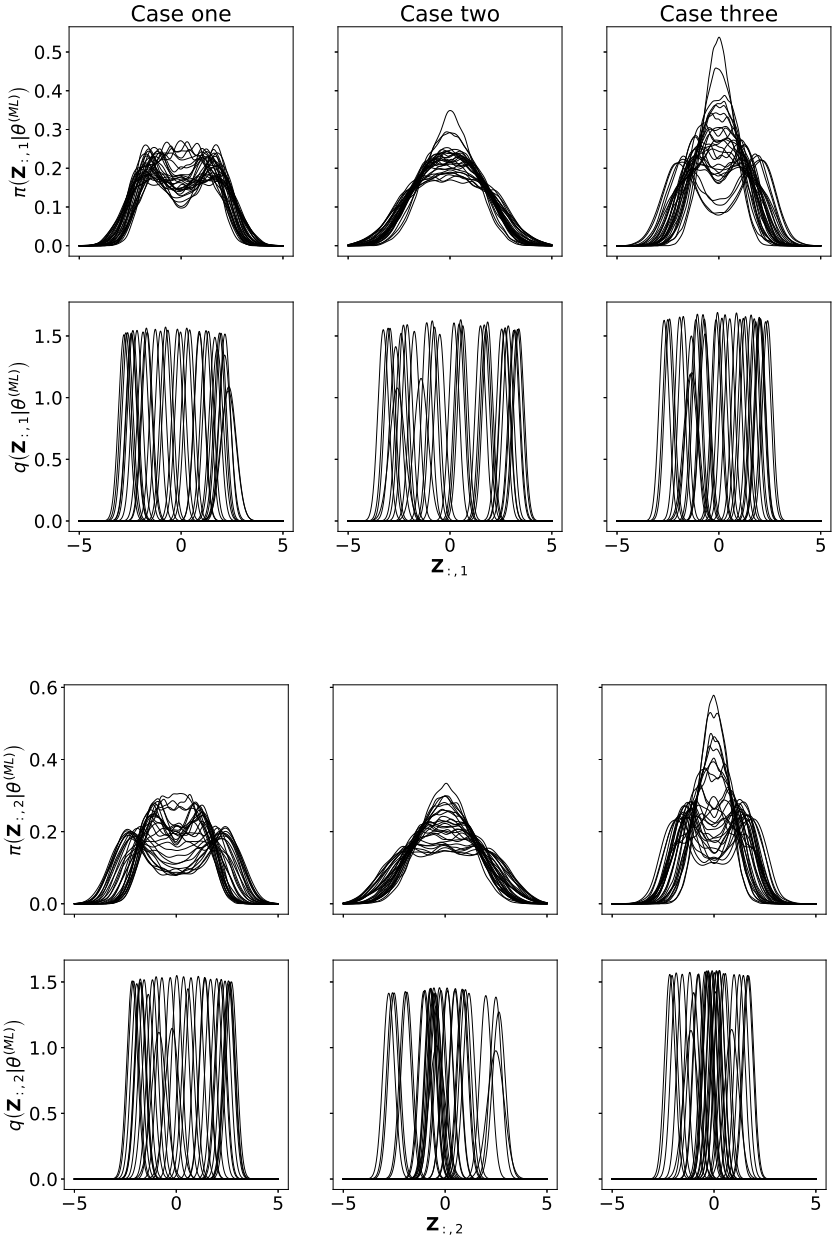


Figure A.8: VEM inference scheme with $k_z = 6$. Marginal latent posterior distributions for all samples, conditional on the approximate maximum marginal likelihood hyperparameters $\theta^{(ML)}$. The exact posterior (first and third row) is obtained using Kernel Density Estimation on the ESS samples, and the variational approximation (second and fourth row) is known analytically. The three columns correspond to the three data generating cases. The first two rows corresponds to the first latent dimension, whilst the last two rows refer to the second latent dimension.

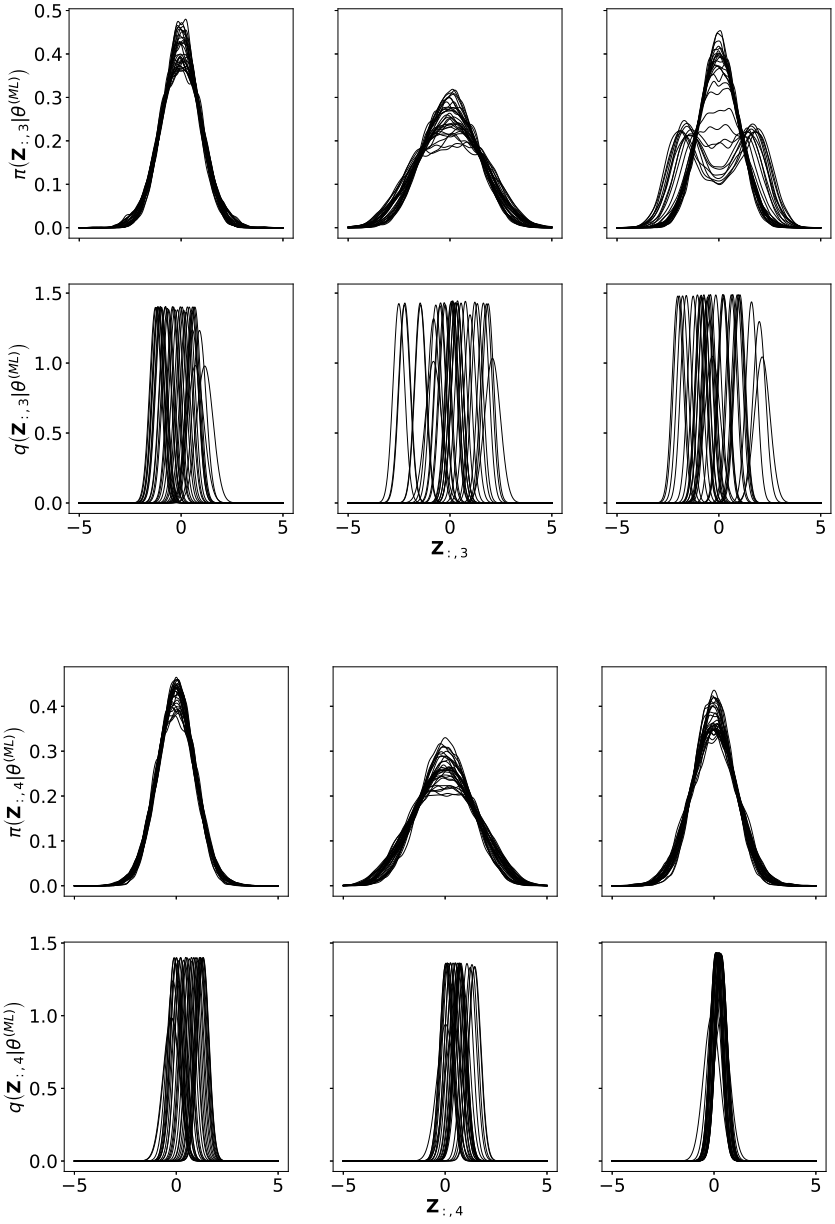


Figure A.9: VEM inference scheme with $k_z = 6$. Marginal latent posterior distributions for all samples, conditional on the approximate maximum marginal likelihood hyperparameters $\theta^{(ML)}$. The exact posterior (first and third row) is obtained using Kernel Density Estimation on the ESS samples, and the variational approximation (second and fourth row) is known analytically. The three columns correspond to the three data generating cases. The first two rows corresponds to the third latent dimension, whilst the last two rows refer to the fourth latent dimension.

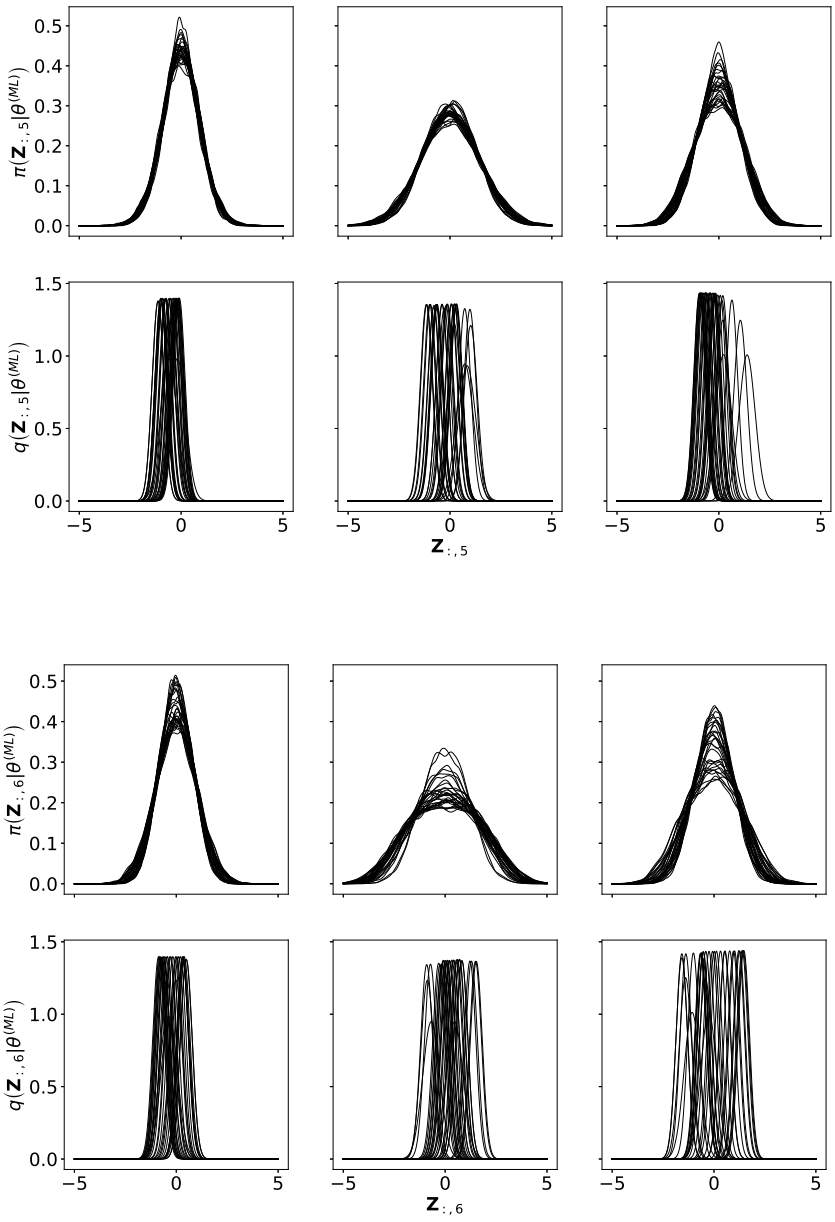


Figure A.10: VEM inference scheme with $k_z = 6$. Marginal latent posterior distributions for all samples, conditional on the approximate maximum marginal likelihood hyperparameters $\theta^{(ML)}$. The exact posterior (first and third row) is obtained using Kernel Density Estimation on the ESS samples, and the variational approximation (second and fourth row) is known analytically. The three columns correspond to the three data generating cases. The first two rows corresponds to the fifth latent dimension, whilst the last two rows refer to the sixth latent dimension.

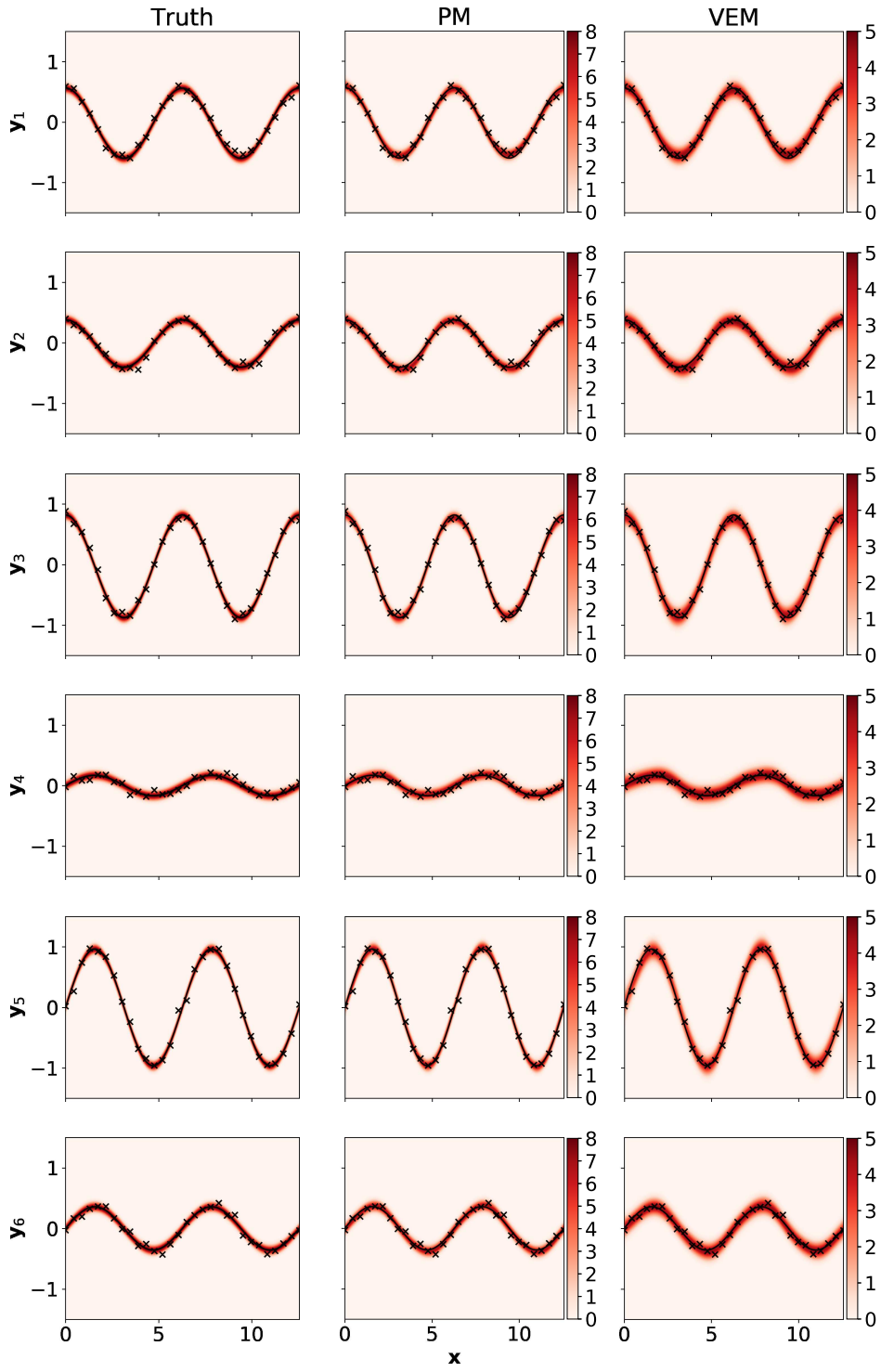


Figure A.11: Case 1 predictive densities. Each row corresponds to a different output dimension. The first column is the true density, the second is the PM approximation and the last is the VEM approximation.

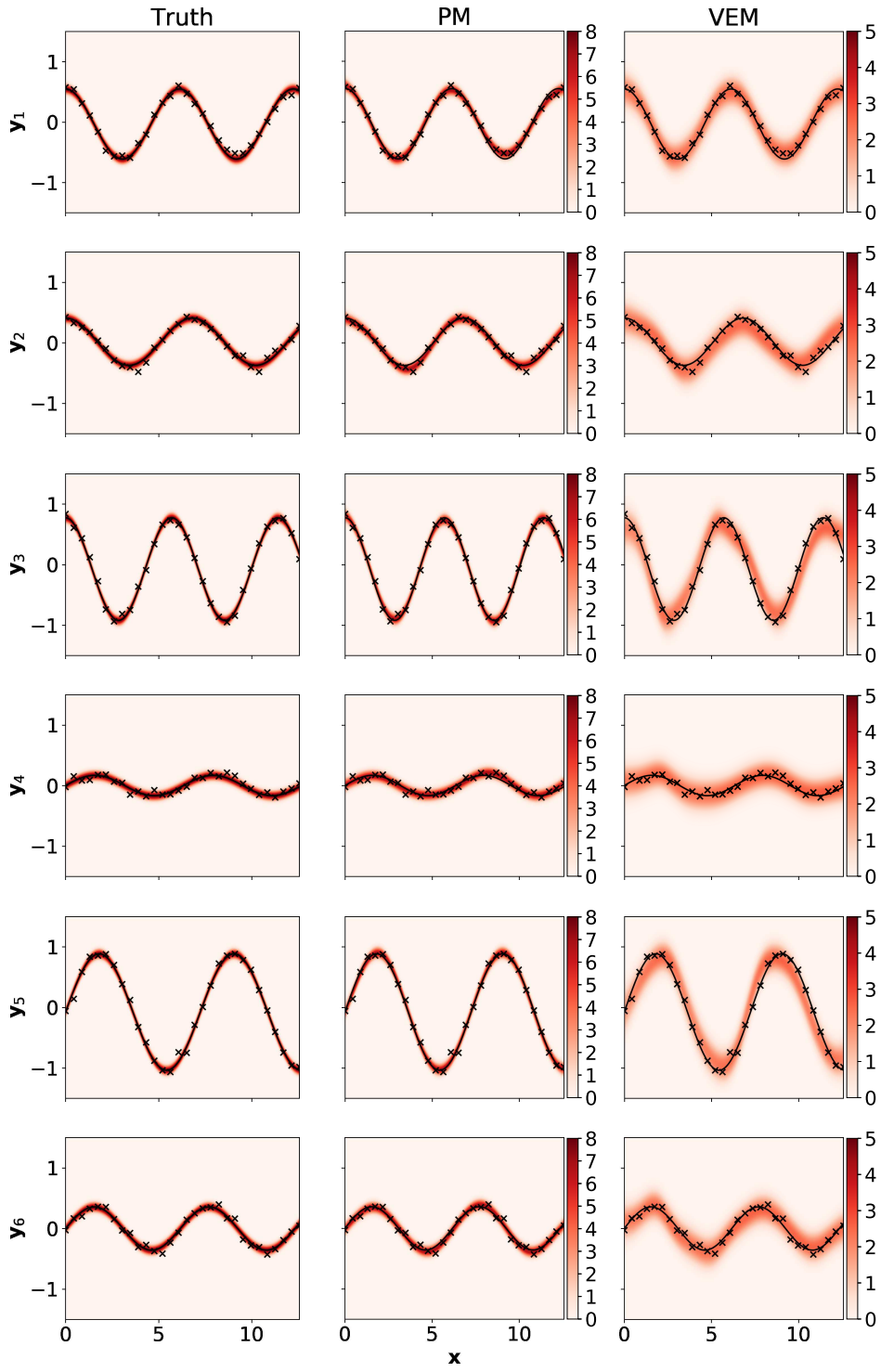


Figure A.12: Case 2 predictive densities. Each row corresponds to a different output dimension. The first column is the true density, the second is the PM approximation and the last is the VEM approximation.

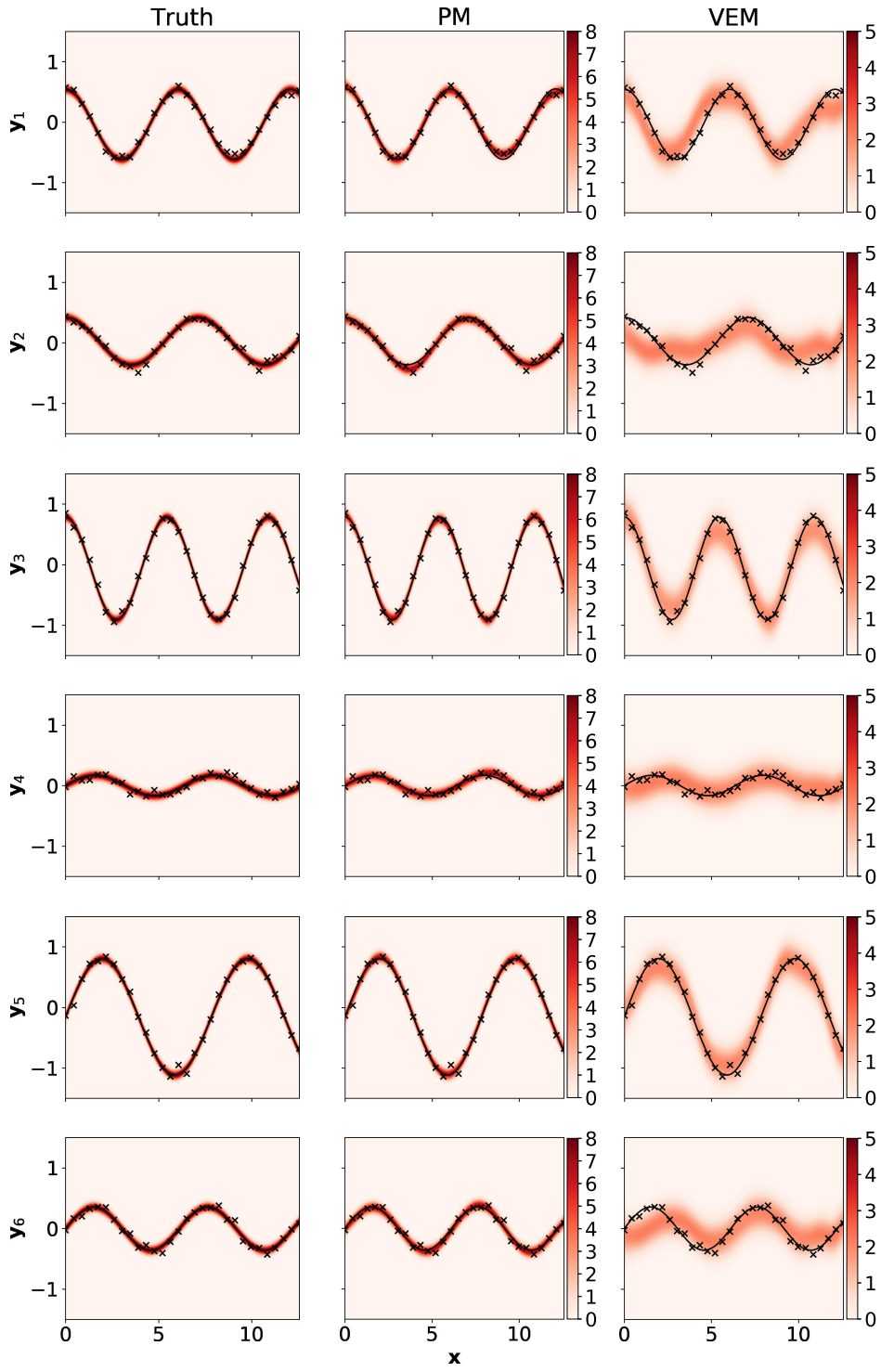


Figure A.13: Case 3 predictive densities. Each row corresponds to a different output dimension. The first column is the true density, the second is the PM approximation and the last is the VEM approximation.

Appendix B

Supplementary material for Chapter 4

B.1 Moments of the marginal distribution over z

Focusing on the i-th feature of $\mathbf{z}$, we find the first two moments, i.e., the mean and variance, of the marginal distribution $p\left(z_i|\mathcal{D}, \boldsymbol{\theta}_i, \beta_i\right)$. Following Girard and Murray-Smith [2003], we approximate $p\left(z_i|\mathcal{D}, \boldsymbol{\theta}_i, \beta_i\right)$ as a Gaussian with mean m and variance v:

$$p\left(z_i|\mathcal{D}, \boldsymbol{\theta}_i, \beta_i\right) = \int p(z_i|\boldsymbol{\xi}', \mathcal{D}, \boldsymbol{\theta}_i, \beta_i) p(\boldsymbol{\xi}') d\boldsymbol{\xi}' \approx \mathcal{N}\left(m, v\right). \tag{B.1}$$

Below we use the notation $\mathbb{E}_\chi[\cdot]$ and $\mathbb{V}\text{ar}_\chi(\cdot)$ to denote an expectation and variance operator with respect to a random variable χ, respectively. Using Fubini's theorem and the laws of total expectation and variance, the moments are then given by:

$$
\begin{aligned}
m &= \int z_i' \, p\left(z_i' | \mathcal{D}, \boldsymbol{\theta}_i, \beta_i\right) dz_i' \\
&= \int z_i' \left[\int p\left(z_i' | \boldsymbol{\xi}', \mathcal{D}, \boldsymbol{\theta}_i, \beta_i\right) p(\boldsymbol{\xi}') d\boldsymbol{\xi}' \right] dz_i' \\
&= \int \left[\int z_i' \, p\left(z_i' | \boldsymbol{\xi}', \mathcal{D}, \boldsymbol{\theta}_i, \beta_i\right) dz_i' \right] p(\boldsymbol{\xi}') d\boldsymbol{\xi}' \\
&= \mathbb{E}_{\boldsymbol{\xi}} \left[\mathbb{E}_{z_i} \left[z_i | \boldsymbol{\xi}, \mathcal{D}, \boldsymbol{\theta}_i, \beta_i \right] \right] \\
&= \mathbb{E}_{\boldsymbol{\xi}} \left[\mu(\boldsymbol{\xi}) \right] \\
&= \mathbb{E}_{\boldsymbol{\xi}} \left[\mathbf{c}_h(\boldsymbol{\xi}, \boldsymbol{\Xi}; \boldsymbol{\theta}_i)^T \left(\mathbf{C}_i + \beta_i^{-1} \mathbf{I} \right)^{-1} \mathbf{z}_{:,i} \right] \\
&= \mathbb{E}_{\boldsymbol{\xi}} \left[\mathbf{c}_h(\boldsymbol{\xi}, \boldsymbol{\Xi}; \boldsymbol{\theta}_i) \right]^T \left(\mathbf{C}_i + \beta_i^{-1} \mathbf{I} \right)^{-1} \mathbf{z}_{:,i}
\end{aligned}
\tag{B.2}
$$

and:

$$
\begin{aligned}
v &= \int (z_i')^2 \, p\left(z_i' | \mathcal{D}, \boldsymbol{\theta}_i, \beta_i\right) dz_i' - m^2 \\
&= \int (z_i')^2 \left[\int p\left(z_i' | \boldsymbol{\xi}', \mathcal{D}, \beta_i\right) p(\boldsymbol{\xi}') d\boldsymbol{\xi}' \right] dz_i' - m^2 \\
&= \mathbb{E}_{\boldsymbol{\xi}} \left[\mathbb{V}\mathrm{ar}_{z_i} \left(z_i | \boldsymbol{\xi}, \mathcal{D}, \boldsymbol{\theta}_i, \beta_i \right) \right] + \mathbb{V}\mathrm{ar}_{\boldsymbol{\xi}} \left(\mathbb{E}_{z_i} \left[z_i | \boldsymbol{\xi}, \mathcal{D}, \boldsymbol{\theta}_i, \beta_i \right] \right) \\
&= \mathbb{E}_{\boldsymbol{\xi}} \left[\sigma^2(\boldsymbol{\xi}) \right] + \mathbb{V}\mathrm{ar}_{\boldsymbol{\xi}} \left(\mu(\boldsymbol{\xi}) \right) \\
&= \mathbb{E}_{\boldsymbol{\xi}} \left[\sigma^2(\boldsymbol{\xi}) \right] + \mathbb{E}_{\boldsymbol{\xi}} \left[\mu(\boldsymbol{\xi})^2 \right] - m^2 \\
&= \mathbb{E}_{\boldsymbol{\xi}} \left[c_h(\boldsymbol{\xi}, \boldsymbol{\xi}; \boldsymbol{\theta}_i) - \mathbf{c}_h(\boldsymbol{\xi}, \boldsymbol{\Xi}; \boldsymbol{\theta}_i)^T \left(\mathbf{C} + \beta_i^{-1} \mathbf{I} \right)^{-1} \mathbf{c}_h(\boldsymbol{\xi}, \boldsymbol{\Xi}; \boldsymbol{\theta}_i) \right] \\
&+ \mathbb{E}_{\boldsymbol{\xi}} \left[\left(\mathbf{c}_h(\boldsymbol{\xi}, \boldsymbol{\Xi}; \boldsymbol{\theta}_i)^T \left(\mathbf{C}_i + \beta_i^{-1} \mathbf{I} \right)^{-1} \mathbf{z}_{:,i} \right)^2 \right] - m^2 \\
&= \mathbb{E}_{\boldsymbol{\xi}} \left[c_h(\boldsymbol{\xi}, \boldsymbol{\xi}; \boldsymbol{\theta}_i) \right] - m^2 \\
&- \left[\left(\mathbf{C}_i + \beta_i^{-1} \mathbf{I} \right)^{-1} - \left(\left(\mathbf{C}_i + \beta_i^{-1} \mathbf{I} \right) \mathbf{z}_{:,i} \right)^2 \right] \mathbb{E}_{\boldsymbol{\xi}} \left[\mathbf{c}_h(\boldsymbol{\xi}, \boldsymbol{\Xi}; \boldsymbol{\theta}_i)^T \mathbf{c}_h(\boldsymbol{\xi}, \boldsymbol{\Xi}; \boldsymbol{\theta}_i) \right].
\end{aligned}
\tag{B.3}
$$

B.2 Kernel expectation

Given a squared exponential kernel and a Gaussian stochastic input distribution, we are able to analytically find the mean and variance of the marginalised latent predictive distribution. This kernel takes the form:

$$c_h(\boldsymbol{\xi}, \boldsymbol{\xi}'; \boldsymbol{\theta}_i) = s \exp\left(-\frac{1}{2}\left(\boldsymbol{\xi} - \boldsymbol{\xi}'\right)^T \mathbf{A}\left(\boldsymbol{\xi} - \boldsymbol{\xi}'\right)\right), \tag{B.4}$$

where $\mathbf{A}$ is a diagonal matrix whose elements are inversely proportional to the correlation lengths across input dimensions. For computational convenience, we write this covariance function in Gaussian function form with normalizing constant $a = (2\pi)^{k_\xi/2} |\mathbf{A}|^{\frac{1}{2}} s$:

$$c_h(\boldsymbol{\xi}, \boldsymbol{\xi}'; \boldsymbol{\theta}_i) = a\mathcal{N}_{\boldsymbol{\xi}}\left(\boldsymbol{\xi}', \mathbf{A}\right). \tag{B.5}$$

where the notation $\mathcal{N}_{\boldsymbol{\chi}}(\cdot, \cdot)$ denotes a normal distribution over a random vector $\boldsymbol{\chi}$, with mean and covariance matrix given by the first and second arguments respectively. We wish to evaluate:

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\xi}}\left[c_h(\boldsymbol{\xi}, \boldsymbol{\xi}; \boldsymbol{\theta}_i)\right] &= a, \\
\mathbb{E}_{\boldsymbol{\xi}}\left[\mathbf{c}_h(\boldsymbol{\xi}, \boldsymbol{\Xi}; \boldsymbol{\theta}_i)\right] &= \mathbb{E}_{\boldsymbol{\xi}}\left[\mathbf{c}_h(\boldsymbol{\xi}, \boldsymbol{\xi}; \boldsymbol{\theta}_i)\right] = a \int \mathcal{N}_{\boldsymbol{\xi}}\left(\boldsymbol{\xi}, \mathbf{A}\right) \mathcal{N}_{\boldsymbol{\xi}}\left(\boldsymbol{\mu}, \Sigma_{\boldsymbol{\xi}}\right) d\boldsymbol{\xi}, \\
\mathbb{E}_{\boldsymbol{\xi}}\left[\mathbf{c}_h(\boldsymbol{\xi}, \boldsymbol{\Xi}; \boldsymbol{\theta}_i)^T \mathbf{c}_h(\boldsymbol{\xi}, \boldsymbol{\Xi}; \boldsymbol{\theta}_i)\right] &= \mathbb{E}_{\boldsymbol{\xi}}\left[\mathbf{c}_h(\boldsymbol{\xi}, \boldsymbol{\xi}; \boldsymbol{\theta}_i) \mathbf{c}_h(\boldsymbol{\xi}, \boldsymbol{\xi}; \boldsymbol{\theta}_i)\right] \\
&= a^2 \int \mathcal{N}_{\boldsymbol{\xi}}\left(\boldsymbol{\xi}, \mathbf{A}\right) \mathcal{N}_{\boldsymbol{\xi}}\left(\boldsymbol{\xi}, \mathbf{A}\right) \mathcal{N}_{\boldsymbol{\xi}}\left(\boldsymbol{\mu}, \Sigma_{\boldsymbol{\xi}}\right) d\boldsymbol{\xi},
\end{aligned} \tag{B.6}$$

in which $(\boldsymbol{\mu}, \Sigma_{\boldsymbol{\xi}})$ are the stochastic input distribution moments. The solutions can be found by using the product of Gaussians rule:

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\xi}}\left[\mathbf{c}_h(\boldsymbol{\xi}, \boldsymbol{\xi}; \boldsymbol{\theta}_i)\right] &= a\mathcal{N}_{\boldsymbol{\mu}}\left(\boldsymbol{\xi}, \mathbf{A} + \Sigma_{\boldsymbol{\xi}}\right), \\
\mathbb{E}_{\boldsymbol{\xi}}\left[\mathbf{c}_h(\boldsymbol{\xi}, \boldsymbol{\xi}; \boldsymbol{\theta}_i) \mathbf{c}_h(\boldsymbol{\xi}, \boldsymbol{\xi}; \boldsymbol{\theta}_i)\right] &= a^2 \mathcal{N}_{\boldsymbol{\xi}}\left(\boldsymbol{\xi}, 2\mathbf{A}\right) \mathcal{N}_{\boldsymbol{\mu}}\left(\boldsymbol{\xi}, \Sigma_{\boldsymbol{\xi}} + \frac{\mathbf{A}}{2}\right).
\end{aligned} \tag{B.7}$$

B.3 Numerical algorithm for Richards equation

Let $\psi_{i',j',k'}^{n',m'}$ denote the value of a quantity ψ at time step n' (time $t = n'\Delta t$ for a constant time step Δt), at Picard iteration m' and at the spatial location $x_1 = i'\Delta x_1$, $x_2 = j'\Delta x_2$ and $x_3 = k'\Delta x_3$. The spatial and temporal discretisations lead to:

$$\begin{aligned} a_1 h_{i-1,j,k}^{n+1,m+1} + b h_{i,j,k}^{n+1,m+1} + c_1 h_{i+1,j,k}^{n+1,m+1} + a_2 h_{i,j-1,k}^{n+1,m+1} + c_2 h_{i,j+1,k}^{n+1,m+1} \\ + a_3 h_{i,j,k-1}^{n+1,m+1} + c_3 h_{i,j,k+1}^{n+1,m+1} = d, \end{aligned} \tag{B.8}$$

which is applicable to all interior nodes (grid points), and where:

$$\begin{aligned} a_1 &= -\frac{k_{i,j,k}^{n+1,m} + k_{i-1,j,k}^{n+1,m}}{2\Delta x_1^2}, \quad a_2 = -\frac{k_{i,j,k}^{n+1,m} + k_{i,j-1,k}^{n+1,m}}{2\Delta x_2^2}, \quad a_3 = -\frac{k_{i,j,k}^{n+1,m} + k_{i,j,k-1}^{n+1,m}}{2\Delta x_3^2} \\ b &= \frac{u_{i,j,k}^{n+1,m}}{\Delta t} + \frac{k_{i+1,j,k}^{n+1,m} + 2k_{i,j,k}^{n+1,m} + k_{i-1,j,k}^{n+1,m}}{2\Delta x_1^2} \\ &\quad + \frac{k_{i,j+1,k}^{n+1,m} + 2k_{i,j,k}^{n+1,m} + k_{i,j-1,k}^{n+1,m}}{2\Delta x_2^2} + \frac{k_{i,j,k+1}^{n+1,m} + 2k_{i,j,k}^{n+1,m} + k_{i,j,k-1}^{n+1,m}}{2\Delta x_3^2} \\ c_1 &= -\frac{k_{i,j,k}^{n+1,m} + k_{i+1,j,k}^{n+1,m}}{2\Delta x_1^2}, \quad c_2 = -\frac{k_{i,j,k}^{n+1,m} + k_{i,j+1,k}^{n+1,m}}{2\Delta x_2^2}, \quad c_3 = -\frac{k_{i,j,k}^{n+1,m} + k_{i,j,k+1}^{n+1,m}}{2\Delta x_3^2} \\ d &= \frac{-k_{i,j,k+1}^{n+1,m} + k_{i,j,k-1}^{n+1,m}}{2\Delta x_3} + u_{i,j,k}^{n+1,m} \frac{h_{i,j,k}^{n}}{\Delta t} \end{aligned} \tag{B.9}$$

The CSC approximation Rathfelder and Abriola [1994] yields $u_{i,j,k}^{n+1,m} = (\theta_{i,j,k}^{n+1,m} - \theta_{i,j,k}^{n})/(h_{i,j,k}^{n+1,m} - h_{i,j,k}^{n})$. In matrix form, the system of equations (B.8) can be written as:

$$\mathbf{A}(\mathbf{h}^{n+1,m})\mathbf{h}^{n+1,m+1} = \mathbf{a}(\mathbf{h}^{n+1,m}) \tag{B.10}$$

in which $\mathbf{h}^{n+1,m'} \in \mathbb{R}^{k_y}$ is a vector of values of $h_{i,j,k}^{n+1,m'}$, $i = 1, \ldots, n_1$, $j = 1, \ldots, n_2$, $k = 1, \ldots, n_3$. $\mathbf{A} \in \mathbb{R}^{k_y \times k_y}$ and $\mathbf{a} \in \mathbb{R}^{k_y}$ depend only on values of the head at iteration m. Thus, the system (B.10) is linear in $\mathbf{h}^{n+1,m+1}$. It can be solved by iterating (in m) within each time step n until convergence; that is, for each time step n, m is incremented until the residual satisfies $||\mathbf{A}(\mathbf{h}^{n+1,m+1})\mathbf{h}^{n+1,m+1} - \mathbf{a}(\mathbf{h}^{n+1,m+1})|| < \varepsilon$ for some specified tolerance ε. In the results presented in section 4.4.6, we use $n_1 = n_2 = n_3 = 26$ ($\Delta x_1 = \Delta x_2 = \Delta x_3 = 0.8$ cm), $\Delta t = 0.5$ s and $\varepsilon = 0.01$.

Appendix C

Supplementary material for Chapter 5

C.1 Generalised Gaussian process experts

Some examples of generalised GP experts include:

Gaussian: with identity link function,

$$p(y|x, \theta_j) = \mathrm{N}(y|m_j(x), \sigma_j^2).$$

Bernoulli:

$$p(y|x, \theta_j) = \mathrm{Bern}(y|g^{-1}(m_j(x))),$$

where the link function maps $(0, 1)$ to the real line, e.g. logistic, probit. For the logistic link function,

$$\mathbb{P}(y = 1|x, \theta_j) = \frac{\exp(m_j(x))}{1 + \exp(m_j(x))}.$$

For the probit link function,

$$\mathbb{P}(y = 1|x, \theta_j) = \Phi(m_j(x)),$$

where Φ denotes the standard normal cumulative distribution function. In this case, the model can be equivalently formulated through a latent response $\tilde{y}$ that is

Gaussian distributed with mean $m_j(x)$ and unit variance. In particular,

$$\tilde{y}|m_j \sim \mathrm{N}(m_j(x), 1) \quad \text{and} \quad p(y|\tilde{y}) = \begin{cases} \mathbf{1}(\tilde{y} \leq 0) & \text{if } l = 0 \\ \mathbf{1}(\tilde{y} > 0) & \text{if } l = 1 \end{cases}.$$

The probit model is recovered by marginalising the latent $\tilde{y}$.

Categorical: with categories $l = 0, \ldots, L$,

$$p(y|x, \theta_j) = \mathrm{Cat}(y|g^{-1}(m_j(x))),$$

where the link function maps the L-dimensional simplex to $\mathbb{R}^L$. For the multivariate logistic link function,

$$\mathbb{P}(y = l|x, \theta_j) = \frac{\exp(m_{j,l}(x))}{1 + \sum_{l=1}^{L} \exp(m_{j,l}(x))} \quad \text{for } l = 1, \ldots, L.$$

For the multinomial probit link function,

$$\mathbb{P}(y = l|x, \theta_j) = \mathbb{P}(\tilde{y}_l > \max(\tilde{y}_1, \ldots, \tilde{y}_{l-1}, \tilde{y}_{l+1}, \ldots, \tilde{y}_L, 0)) \quad \text{for } l = 1, \ldots, L,$$

where $\tilde{y}$ takes values in $\mathbb{R}^L$ has multivariate Gaussian distribution with mean $m_j(x) = (m_{j,1}(x), \ldots, m_{j,L}(x))^T$ and covariance matrix Σ_j, which may be the identity matrix, or treated as a more general scale parameter (in this case, care should be taken to avoid identifiability issues). The prior on the vector-valued unknown function $m_j(x)$ can be extended to independent GPs across $l = 1, \ldots, L$ or a matrix-variate GP.

Ordinal: with ordered categories $l = 0, \ldots, L$ and cutoffs $0 = \varepsilon_0 < \varepsilon_1 < \ldots < \varepsilon_{L-1}$,

$$\mathbb{P}(y \leq l|x, \theta_j) = g^{-1}(\varepsilon_l - m_j(x)),$$

where the link function maps $(0, 1)$ to the real line. Due to the nonparametric nature of the model, we consider fixed cutoffs $\varepsilon_1, \ldots, \varepsilon_{L-1}$. For the logistic link function,

$$\mathbb{P}(y \leq l|x, \theta_j) = \frac{\exp(\varepsilon_l - m_j(x))}{1 + \exp(\varepsilon_l - m_j(x))}.$$

For the probit link function,

$$\mathbb{P}(y \leq l|x, \theta_j) = \Phi\left(\frac{\varepsilon_l - m_j(x)}{\sigma_j}\right),$$

with additional scale parameter σ_j^2. In this case, the model can be equivalently formulated through a latent response $\tilde{y}$ that is Gaussian distributed with mean $m_j(x)$ and variance σ_j^2. In particular,

$$\tilde{y}|m_j, \sigma_j^2 \sim \mathrm{N}(m_j(x), \sigma_j^2) \quad \text{and} \quad p(y|\tilde{y}) = \begin{cases} \mathbf{1}(\tilde{y} \leq 0) & \text{if } l = 0 \\ \mathbf{1}(\varepsilon_{l-1} < \tilde{y} \leq \varepsilon_l) & \text{if } l = 1, \ldots, L-1 \\ \mathbf{1}(\tilde{y} > \varepsilon_{L-1}) & \text{if } l = L \end{cases}.$$

The ordered probit model is recovered by marginalising the latent $\tilde{y}$.

Poisson:

$$p(y|x, \theta_j) = \mathrm{Pois}(y|g^{-1}(m_j(x))),$$

where the link function maps $(0, \infty)$ to $\mathbb{R}$. For the log link function with $\lambda_j(x) = \exp(m_j(x))$,

$$\mathbb{P}(y = l|x, \theta_j) = \frac{\exp(-\lambda_j(x))\lambda_j(x)^l}{l!} \quad \text{for } l = 0, 1, 2, \ldots.$$

C.2 Local input models

Other types of inputs can be easily handled through the assumption of local independence

$$p(x|\psi) = \prod_{p=1}^{P} p(x_p|\psi_p), \tag{C.1}$$

and through the assumption that each parametric model $p(x_p|\psi_p)$ belongs to the exponential family, that is,

$$p(x_p|\psi_p) = \exp(\psi_p' t_p(x_p) - a_p(\psi_p) + b_p(x_p)).$$

The parameter ψ has the standard conjugate prior, which assumes independence ψ_p across $p = 1, \ldots, P$ with

$$\pi(\psi_p) = \exp(\psi_p' \tau_p - \nu_p a_p(\psi_p) + c_p(\tau_p, \nu_p)).$$

In this conjugate setting, the parameters ψ can be marginalised and the marginal likelihood of the inputs in each cluster is available analytically. Specifically, for the collapsed Gibbs sampler, we need 1) the marginal likelihood $h(x_n)$ and 2) the predictive likelihood $h(x_n|\mathbf{X}_j^{-n})$, where $\mathbf{X}_j^{-n}$ contains $x_{n'}$ such that $n' \neq n, z_{n'} = j$. We note that due to the assumption of local independence:

$$h(x_n) = \int p(x_n|\psi)\pi(\psi)d\psi = \prod_{p=1}^{P} h(x_{n,p}),$$

$$h(x_n|\mathbf{X}_j^{-n}) = \int p(x_n|\psi)\pi(\psi|\mathbf{X}_j^{-n})d\psi = \prod_{p=1}^{P} h(x_{n,p}|\mathbf{X}_{j,p}^{-n}).$$

Examples (used in this paper) include:

Gaussian: for continuous input $x_{n,p}$ taking values in $\mathbb{R}$ with

$$p(x_{n,p}|\psi_p) = \mathrm{N}(x_{n,p}|u_p, s_p^2),$$

where $\psi_p = (u_p, s_p^2)$. The standard conjugate prior is the normal-inverse gamma distribution,

$$u_p|s_p^2 \overset{ind}{\sim} \mathrm{N}(u_{0,p}, c_p^{-1}s_p^2), \quad s_p^2 \overset{ind}{\sim} \mathrm{IG}(a_{x,p}, b_{x,p}),$$

which we denote by $(u_p, s_p^2) \overset{ind}{\sim} \text{NIG}(u_{0,p}, c_p, a_{x,p}, b_{x,p})$. In this case, marginally $x_{n,p}$ has a non-central t-distribution,

$$h(x_{n,p}) = t\left(x_{n,p}|u_{0,p}, \frac{b_{x,p}}{a_{x,p}}\frac{c_p+1}{c_p}, 2a_{x,p}\right).$$

The predictive distribution of $x_{n,p}$ given $z_n = j$ is a non-central t-distribution,

$$h(x_{n,p}|\mathbf{X}_{j,p}^{-n}) = t\left(x_{n,p}|\widehat{u}_{j,p}^{-n}, \frac{\widehat{b}_{x,j,p}^{-n}}{\widehat{a}_{x,j,p}^{-n}}\frac{\widehat{c}_{j,p}^{-n}+1}{\widehat{c}_{j,p}^{-n}}, 2\widehat{a}_{x,j,p}^{-n}\right),$$

with $\widehat{c}_{j,p}^{-n} = c_p + N_j^{-n}$, $\widehat{a}_{x,j,p}^{-n} = a_{x,p} + N_j^{-n}/2$,

$$\widehat{u}_{j,p}^{-n} = \frac{1}{c_p + N_j^{-n}}(c_p u_{0,p} + N_j^{-n}\bar{x}_{j,p}^{-n}),$$

$$\widehat{b}_{x,j,p}^{-n} = b_{x,p} + \frac{1}{2}\left(c_p u_{0,p}^2 - \widehat{c}_{j,p}^{-n}(\widehat{u}_{j,p}^{-n})^2 + \sum_{n'\neq n: z_{n'}=j} x_{n',p}^2\right),$$

and $\bar{x}_{j,p}^{-n} = 1/N_j^{-n}\sum_{n'\neq n: z_{n'}=j} x_{n',p}$.

Categorical: for discrete inputs $x_{n,p}$ taking *unordered* values $g = 0, 1, \ldots, G_p$ with

$$p(x_{n,p}|\psi_p) = \psi_{p,x_{n,p}},$$

where ψ_p is $G_p + 1$ vector of probabilities such that $\sum_{g=0}^{G_p}\psi_{p,g} = 1$. The standard conjugate prior is the Dirichlet distribution with parameter $\gamma_p = (\gamma_{p,0}, \ldots, \gamma_{p,G_p})$. In this case, the marginal likelihood is the Dirichlet-multinomial with

$$h(x_{n,p}) = \frac{\Gamma\left(\sum_{g=0}^{G_p}\gamma_{p,g}\right)}{\Gamma\left(\sum_{g=0}^{G_p}\gamma_{p,g}+1\right)}\frac{\Gamma\left(\gamma_{p,x_{n,p}}+1\right)}{\Gamma\left(\gamma_{p,x_{n,p}}\right)}.$$

The predictive likelihood of $x_{n,p}$ given $z_n = j$ is the Dirichlet-multinomial with

$$h(x_{n,p}|\mathbf{X}_{j,p}^{-n}) = \frac{\Gamma\left(\sum_{g=0}^{G_p}\gamma_{p,g}+N_j^{-n}\right)}{\Gamma\left(\sum_{g=0}^{G_p}\gamma_{p,g}+N_j^{-n}+1\right)}\frac{\Gamma\left(\gamma_{p,x_{n,p}}+N_{j,x_{n,p}}^{p,-n}+1\right)}{\Gamma\left(\gamma_{p,x_{n,p}}+N_{j,x_{n,p}}^{p,-n}\right)},$$

where $N_{j,g}^{p,-n} = \sum_{n'\neq n: z_{n'}=j}\mathbf{1}(x_{n',p}=g)$.

Binomial: for discrete inputs $x_{n,p}$ taking *ordered* values $g = 0, 1, \ldots, G_p$ with

$$p(x_{n,p}|\psi_p) = \left(\begin{array}{c} G_p \\ x_{n,p} \end{array} \right) \psi_p^{x_{n,p}} (1-\psi_p)^{G_p - x_{n,p}},$$

where $\psi_p \in (0,1)$. The standard conjugate prior is the beta distribution with parameter $\gamma_p = (\gamma_{p,0}, \gamma_{p,1})$. In this case, the marginal likelihood is the beta-binomial with

$$h(x_{n,p}) = \left(\begin{array}{c} G_p \\ x_{n,p} \end{array} \right) \frac{\Gamma(\gamma_{p,0} + \gamma_{p,1})}{\Gamma(\gamma_{p,0})\,\Gamma(\gamma_{p,1})} \frac{\Gamma(\gamma_{p,0} + x_{n,p})\,\Gamma(\gamma_{p,1} + G_p - x_{n,p})}{\Gamma(\gamma_{p,0} + \gamma_{p,1} + G_p)}.$$

The predictive likelihood of $x_{n,p}$ given $z_n = j$ is the beta-binomial with

$$h(x_{n,p}|\mathbf{X}_{j,p}^{-n}) = \left(\begin{array}{c} G_p \\ x_{n,p} \end{array} \right) \frac{\Gamma\left(\gamma_{p,0} + \gamma_{p,1} + G_p N_j^{-n}\right)}{\Gamma(\widehat{\gamma}_{p,0,j})\,\Gamma(\widehat{\gamma}_{p,1,j})} \frac{\Gamma(\widehat{\gamma}_{p,0,j} + x_{n,p})\,\Gamma(\widehat{\gamma}_{p,1,j} + G_p - x_{n,p})}{\Gamma\left(\gamma_{p,0} + \gamma_{p,1} + G_p(N_j^{-n} + 1)\right)},$$

where $\widehat{\gamma}_{p,0,j} = \gamma_{p,0} + N_j^{-n} \bar{x}_{p,j}^{-n}$ and $\widehat{\gamma}_{p,1,j} = \gamma_{p,1} + N_j^{-n}(G_p - \bar{x}_{p,j}^{-n})$.

C.3 Gibbs sampling for the joint mixture of generalised GP experts

We present the algorithm for a general setting, when the observed outputs y_n are a deterministic function of latent Gaussian outputs $\tilde{y}_n$. This includes the probit, ordered probit and multinomial probit, as well as the Gaussian example with $y = \tilde{y}$. The MCMC algorithm targets the posterior

$$\begin{aligned}\pi(z_{1:N}, \sigma^2_{1:k}, &\beta_{0,1:k}, \lambda_{1:k}, \alpha, \tilde{y}_{1:N} \mid y_{1:N}, x_{1:N}) \\ &\propto \prod_{j=1}^{k} h(\tilde{\boldsymbol{Y}}_j | \sigma_j^2, \beta_{0,j}, \lambda_j) h(\boldsymbol{X}_j) \prod_{j=1}^{k} \pi(\sigma_j^2)\pi(\beta_{0,j})\pi(\lambda_j) \\ &* \frac{\Gamma(\alpha)}{\Gamma(\alpha+N)} \alpha^k \prod_{j=1}^{k} \Gamma(N_j)\pi(\alpha) \prod_{n=1}^{N} p(y_n|\tilde{y}_n),\end{aligned}$$

where we make use of the notation $\boldsymbol{X}_j$ to denote the inputs x_n such that $z_n = j$ and $\tilde{\boldsymbol{Y}}_j$ to denote the latent outputs $\tilde{y}$ such that $z_n = j$. The marginal likelihood of $\mathbf{Y}_j$ given $\beta_{0,j}$, λ_j and σ_j^2, obtained from marginalising the unknown functions m_j, is Gaussian, e.g. for the ordered probit,

$$h(\tilde{\boldsymbol{Y}}_j | \sigma_j^2, \beta_{0,j}, \lambda_j) = \mathrm{N}(\tilde{\boldsymbol{Y}}_j \mid \beta_{0,j} 1_{N_j}, \sigma_j^2 I_{N_j} + K_{\lambda_j}),$$

where K_{λ_j} denotes the N_j by N_j matrix of the kernel function evaluated at every pair of inputs in cluster j. The marginal likelihood of $\boldsymbol{X}_j$, obtained from marginalising ψ_j, is also available in closed form with

$$h(\boldsymbol{X}_j) = \prod_{p=1}^{P} h(\boldsymbol{X}_{j,p}) = \prod_{p=1}^{P} \int \prod_{n:z_n=j} p(x_{n,p}|\psi_p)\pi(\psi_p) d\psi_p.$$

The term $p(y_n|\tilde{y}_n)$ represents the deterministic function specifying the observed output y_n given the latent Gaussian output $\tilde{y}_n$; examples are provided in Appendix 5A.

The algorithm is a Gibbs sampler, which alternatively samples each set of parameters, 1) the allocation variables $z_{1:N}$, 2) the unique cluster parameters $(\sigma_j^2, \beta_{0,j}, \lambda_j)_{j=1}^k$, 3) the mass parameter α and 4) the latent outputs $\tilde{y}_{1:N}$ (if needed).

Allocation variables. A non-conjugate collapsed Gibbs sampler is employed, combining Algorithm 3, when cluster parameters can be integrated, and Algorithm 8, when cluster parameters cannot be integrated, of Neal [2000]. This consists of N

Gibbs steps, where the allocation variable z_n for each data point is updated conditioned on all others $z_1, \ldots, z_{n-1}, z_{n+1}, \ldots, z_N$ through the following steps. Throughout, we make use of the superscript notation $-n$ to denote the data points, parameters and latent variables with the n^{th} data point removed.

1. Remove singleton cluster: If $z_n \neq z_{n'}$ for all $n' \neq n$, i.e. data point n is in a singleton cluster, remove that cluster and set $(\sigma^2_{k^{-n}+1}, \beta_{0,k^{-n}+1}, \lambda_{k^{-n}+1})$ equal to the values of the singleton cluster parameters.

2. Calculate the allocation probability for each occupied cluster: $j \in \{1, \ldots, k^{-n}\}$

$$\begin{aligned} &p(z_n = j | z^{-n}_{1:N}, \sigma^2_j, \lambda_j, \beta_{0,j}, \alpha, x_{1:N}, \tilde{y}_{1:N}) \\ &\quad \propto N_j^{-n} h(\tilde{y}_n | \tilde{\boldsymbol{Y}}_j^{-n}, \sigma^2_j, \lambda_j, \beta_{0,j}) h(x_n | \boldsymbol{X}_j^{-n}), \end{aligned}$$

   where marginal likelihood of $\tilde{y}_n$ conditioned on the latent outputs in cluster j is the predictive density from the GP regression model [Rasmussen and Williams, 2005, Chp. 2].

3. Calculate the allocation probability for m new clusters: sample m new parameters (or $m-1$ new parameters if z_n was in a singleton cluster) from the prior

$$\sigma^2_{k^{-n}+j} \sim \pi(\sigma^2), \quad \beta_{0,k^{-n}+j} \sim \pi(\beta_0), \quad \lambda_{k^{-n}+j} \sim \pi(\lambda).$$

   Then, for $j = 1, \ldots, m$, compute

$$\begin{aligned} &p(z_n = k^{-n} + j | \sigma^2_{k^{-n}+j}, \beta_{0,k^{-n}+j}, \lambda_{k^{-n}+j}, \alpha, \tilde{y}_n, x_n) \\ &\quad \propto \frac{\alpha}{m} h(\tilde{y}_n | \sigma^2_{k^{-n}+j}, \beta_{0,k^{-n}+j}, \lambda_{k^{-n}+j}) h(x_n). \end{aligned}$$

4. Update the allocation variable z_n using the allocation probabilities. All empty clusters are removed, and if one of the m new clusters is selected, then set $z_n = k^{-n}+1$ and the parameters $(\sigma^2_{k^{-n}+1}, \beta_{0,k^{-n}+1}, \lambda_{k^{-n}+1})$ equal to the parameters of the selected new cluster.

Cluster parameters. The parameters for each cluster are conditionally independent across $j = 1, \ldots, k$ with full conditional

$$\pi(\sigma^2_j, \beta_{0,j}, \lambda_j | \tilde{\boldsymbol{Y}}_j) \propto h(\tilde{\boldsymbol{Y}}_j | \sigma^2_j, \beta_{0,j}, \lambda_j) \pi(\sigma^2_j) \pi(\beta_{0,j}) \pi(\lambda_j),$$

which is not available in closed form. We use HMC [Duane et al., 1987] to sample from the full conditional.

Mass parameter. The mass parameter α is updated using the auxiliary variable technique of Escobar and West [1995]. By sampling an auxiliary variable $\xi \sim \text{Beta}(\alpha + 1, N)$; setting

$$\widehat{v}_\alpha = v_\alpha - \log(\xi) \quad \text{and} \quad \widehat{u}_\alpha = \begin{cases} u_\alpha + k - 1 & \text{w/ prob } \frac{N\widehat{v}_\alpha}{N\widehat{v}_\alpha + u_\alpha + k - 1} \\ u_\alpha + k & \text{w/ prob } \frac{u_\alpha + k - 1}{N\widehat{v}_\alpha + u_\alpha + k - 1} \end{cases} ;$$

and sampling $\alpha \sim \text{Ga}(\widehat{u}_\alpha, \widehat{v}_\alpha)$.

Latent outputs. The latent outputs are independent across cluster $j = 1, \ldots, k$, with full conditional

$$\pi(\tilde{\boldsymbol{Y}}_j | \boldsymbol{Y}_j, \sigma_j^2, \beta_{0,j}, \lambda_j) \propto h(\tilde{\boldsymbol{Y}}_j | \sigma_j^2, \beta_{0,j}, \lambda_j) \prod_{n:z_n=j} p(y_n | \tilde{y}_n).$$

In the Gaussian case, $p(y_n|\tilde{y}_n) = \mathbf{1}(y_n = \tilde{y}_n)$, and this step is not needed. For the other probit-type models, the full conditional of the latent outputs in cluster j is a truncated multivariate Gaussian, which is sampled through a Gibbs algorithm combined with cumulative distribution function inversion techniques [Kotecha and Djuric, 1999].

C.4 Predictions for the joint mixture of generalised GP experts

In the Gaussian example, the posterior density for a new output y_* given a new x_* is given by

$$f(y_*|x_*, y_{1:N}, x_{1:N}) = \int f(y_*|x_*, y_{1:N}, x_{1:N}, \zeta) \frac{\pi(\zeta|y_{1:N}, x_{1:N}) f(x_*|x_{1:N}, \zeta)}{f(x_*|x_{1:N})} d\zeta$$
$$\approx C^{-1} \left(\sum_{m=1}^{M} p^{(m)}_{k^{(m)}+1}(x_*) h(y_*) + \sum_{j=1}^{k^{(m)}} p^{(m)}_j(x_*) h(y_*|\boldsymbol{Y}^{(m)}_j, \beta^{(m)}_{0,j}, \lambda^{(m)}_j, \sigma^{2\,(m)}_j) \right). \tag{C.2}$$

In this case, we have a weighted average of the GP predictive densities across clusters and the marginal likelihood $h(y_*)$ for a new cluster. Note that the marginal likelihood $h(y_*)$ for a new cluster is unavailable in closed form as it requires integration over the parameters $(\beta_0, \lambda, \sigma^2)$. However, we can compute a simple Monte Carlo estimate of this quantity by sampling from the prior,

$$h(y_*) \approx \frac{1}{S} \sum_{s=1}^{S} \mathrm{N}(y_* \mid \beta_0^s, \sigma^{2\,s} + K_{\lambda^s}(x_*, x_*)),$$

with $(\sigma^{2\,s}, \beta_0^s, \lambda^s)$ i.i.d. samples from the prior. For other types of outputs through probit models, we can similarly use the MCMC output to compute predictive quantities of interest at a test input x_*.

The ordered probit with ordered categories $l = 0, \ldots, L$ and fixed cutoffs $0 = \varepsilon_0 < \varepsilon_1 < \ldots < \varepsilon_{L-1}$. First note, that we can compute the expectation and density of the latent continuous $\tilde{y}_*$ given a test input x_*, as in the Gaussian example. The posterior probability that $y_* = l$ given a test input x_* is

$$\mathbb{P}(y_* = l|x_*, y_{1:N}, x_{1:N}) = \int \mathbb{P}(y_* = l|x_*, y_{1:N}, x_{1:N}, \zeta) \frac{\pi(\zeta|y_{1:N}, x_{1:N}) f(x_*|x_{1:N}, \zeta)}{f(x_*|x_{1:N})} d\zeta$$
$$\approx C^{-1} \left(\sum_{m=1}^{M} p^{(m)}_{k^{(m)}+1}(x_*) \mathbb{P}(y_* = l|x_*) + \sum_{j=1}^{k^{(m)}} p^{(m)}_j(x_*) \mathbb{P}(y_* = l|x_*, \tilde{\boldsymbol{Y}}^{(m)}_j, \sigma^{2\,(m)}_j, \beta^{(m)}_{0,j}, \lambda^{(m)}_j) \right).$$

For cluster j of sample m, the probability that $y_* = l$ is

$$\mathbb{P}(y_* = l|x_*, \tilde{\boldsymbol{Y}}_j^{(m)}, \sigma_j^{2\,(m)}, \beta_{0,j}^{(m)}, \lambda_j^{(m)}) = \mathbb{P}(\varepsilon_{l-1} < \tilde{y}_* \leq \varepsilon_l|x_*, \tilde{\boldsymbol{Y}}_j^{(m)}, \sigma_j^{2\,(m)}, \beta_{0,j}^{(m)}, \lambda_j^{(m)})$$
$$= \Phi\left(\frac{\varepsilon_l - \widehat{m}_j^{(m)}(x_*)}{\sqrt{\widehat{K}_j^{(m)}(x_*, x_*) + \sigma_j^{2\,(m)}}}\right) - \Phi\left(\frac{\varepsilon_{l-1} - \widehat{m}_j^{(m)}(x_*)}{\sqrt{\widehat{K}_j^{(m)}(x_*, x_*) + \sigma_j^{2\,(m)}}}\right),$$

with $\varepsilon_{-1} = -\infty$, $\varepsilon_L = \infty$ and $\widehat{m}_j^{(m)}(x_*)$ and $\widehat{K}_j^{(m)}(x_*, x_*)$ denoting the GP predictive mean and kernel functions in cluster j of sample m. For a new cluster, the marginal probability $\mathbb{P}(y_* = l|x_*)$ is unavailable in closed form as it requires integration over the parameters $(\beta_0, \lambda, \sigma^2)$. We can again employ a Monte Carlo approach to estimate this quantity,

$$\mathbb{P}(y_* = l|x_*) \approx \frac{1}{S}\sum_{s=1}^{S} \Phi\left(\frac{\varepsilon_l - \beta_0^s}{\sqrt{K_{\lambda^s}(x_*, x_*)) + \sigma^{2\,s}}}\right) - \Phi\left(\frac{\varepsilon_{l-1} - \beta_0^s}{\sqrt{K_{\lambda^s}(x_*, x_*)) + \sigma^{2\,s}}}\right),$$

with $(\sigma^{2\,s}, \beta_0^s, \lambda^s)$ i.i.d. samples from the prior.

An advantage of jointly modelling the outputs and inputs includes the possibility to compute the predictive distribution of y_* based only on a subset of inputs, say only based on a single input x_{*p}. In this case, the weights would only involve the local predictive marginal likelihood of x_{*p} for each cluster $h(x_{*p}|\mathbf{X}_{j,p}^{(m)})$, $j = 1, \ldots, k^{(m)}$, and for a new cluster $h(x_{*p})$. However, the local expectation would need to be integrated with respect to local predictive marginal likelihood of $x_{*-p} = (x_{*1}, \ldots, x_{*p-1}, x_{*p+1}, \ldots, x_{*P})$ in each cluster. For example, in the Gaussian case,

$$\mathbb{E}[y_*|x_{*p}, y_{1:N}, x_{1:N}]$$
$$\approx C_p^{-1}\left(\sum_{m=1}^{M} p_{k^{(m)}+1}^{(m)}(x_{*p})\mu_\beta + \sum_{j=1}^{k^{(m)}} p_j^{(m)}(x_{*p})\left(\int \widehat{m}_j^{(m)}(x_*)\prod_{p'\neq p} h(x_{*p'}|\mathbf{X}_{j,p'}^{(m)})dx_{*-p}\right)\right),$$

with

$$p_{k^{(m)}+1}^{(m)}(x_{*p}) = \frac{\alpha^{(m)}}{\alpha^{(m)} + N}h(x_{*p}), \quad p_j^{(m)}(x_{*p}) = \frac{N_j^{(m)}}{\alpha^{(m)} + N}h(x_{*p}|\mathbf{X}_{j,p}^{(m)}),$$

and

$$C_p = \sum_{m=1}^{M} \frac{\alpha^{(m)}}{\alpha^{(m)} + N}h(x_{*p}) + \sum_{j=1}^{k^{(m)}} \frac{N_j^{(m)}}{\alpha^{(m)} + N}h(x_{*p}|\mathbf{X}_{j,p}^{(m)}).$$

C.5 Gibbs sampling for the enriched mixture of generalised GP experts

The MCMC algorithm targets the posterior

$$\pi(z_{1:N}, \sigma^2_{1:k}, \beta_{0,1:k}, \lambda_{1:k}, \alpha_\theta, \alpha_{\psi,1:k}, \tilde{y}_{1:N} \mid y_{1:N}, x_{1:N}) \propto \prod_{j=1}^{k} h(\tilde{\boldsymbol{Y}}_j | \sigma^2_j, \beta_{0,j}, \lambda_j) \prod_{l=1}^{k_j} h(\boldsymbol{X}_{l|j})$$
$$* \prod_{n=1}^{N} p(y_n|\tilde{y}_n) \frac{\Gamma(\alpha_\theta)}{\Gamma(\alpha_\theta + N)} \alpha^k_\theta \pi(\alpha_\theta) \prod_{j=1}^{k} \alpha^{k_j}_{\psi,j} \frac{\Gamma(\alpha_{\psi,j})\Gamma(N_j)}{\Gamma(\alpha_{\psi,j} + N_j)} \pi(\sigma^2_j)\pi(\beta_{0,j})\pi(\lambda_j)\pi(\alpha_{\psi,j}) \prod_{l=1}^{k_j} \Gamma(N_{l|j}),$$

where $\mathbf{X}_{l|j} = (x_n)_{z_n=(j,l)}$. The algorithm proceeds as described in Appendix 5C, with changes to sample the nested allocation variables $z_{1:N}$ and the additional mass parameters $\alpha_{\psi,j}$. These changes are detailed below.

Allocation variables. The non-conjugate collapsed Gibbs sampler is extended to sample the bivariate allocation variable $z_n = (z_{y,n}, z_{x,n})$ for each data point conditioned on all others $z_1, \ldots, z_{n-1}, z_{n+1}, \ldots, z_N$ through the following steps.

1. Remove singleton cluster:

   - Singleton y-cluster: If $z_{y,n} \neq z_{y,n'}$ for all $n' \neq n$, i.e. data point n is in a singleton y-cluster, remove that cluster and set $(\sigma^2_{k^{-n}+1}, \beta_{0,k^{-n}+1}, \lambda_{k^{-n}+1}, \alpha_{\psi,k^{-n}+1})$ equal to the values of the singleton cluster parameters.
   - Singleton x-cluster within a non-singleton y-cluster: If $z_{y,n} = z_{y,n'}$ for some $n' \neq n$ and $z_{x,n} \neq z_{x,n'}$ for all $n' \neq n$ such that $z_{y,n} = z_{y,n'}$, i.e. data point n is in a singleton x-cluster within a non-singleton y-cluster, remove that cluster.

2. Calculate the allocation probability for each occupied cluster: $j \in \{1, \ldots, k^{-n}\}$ and $l \in \{1, \ldots, k^{-n}_j\}$

$$p(z_n = (j,l) | z^{-n}_{1:N}, \sigma^2_j, \lambda_j, \beta_{0,j}, \alpha_\theta, \alpha_{\psi,j}, x_{1:N}, \tilde{y}_{1:N})$$
$$\propto \frac{N^{-n}_j N^{-n}_{l|j}}{\alpha_{\psi,j} + N^{-n}_j} h(\tilde{y}_n | \tilde{\boldsymbol{Y}}^{-n}_j, \sigma^2_j, \lambda_j, \beta_{0,j}) h(x_n | \mathbf{X}^{-n}_{l|j}).$$

3. Calculate the allocation probability for a new x-cluster within each occupied

y-cluster: $j \in \{1, \ldots, k^{-n}\}$

$$p(z_n = (j, k_j^{-n} + 1) | z_{1:N}^{-n}, \sigma_j^2, \lambda_j, \beta_{0,j}, \alpha_\theta, \alpha_{\psi,j}, x_{1:N}, \tilde{y}_{1:N})$$
$$\propto \frac{N_j^{-n} \alpha_{\psi,j}}{\alpha_{\psi,j} + N_j^{-n}} h(\tilde{y}_n | \tilde{\boldsymbol{Y}}_j^{-n}, \sigma_j^2, \lambda_j, \beta_{0,j}) h(x_n).$$

4. Calculate the allocation probability for m new y-clusters: sample m new parameters (or $m-1$ new parameters if $z_{y,n}$ was in a singleton y-cluster) from the prior

$$\sigma^2_{k^{-n}+j} \sim \pi(\sigma^2), \quad \beta_{0,k^{-n}+j} \sim \pi(\beta_0), \quad \lambda_{k^{-n}+j} \sim \pi(\lambda), \quad \alpha_{\psi,k^{-n}+j} \sim \text{Ga}(u_\psi, v_\psi).$$

Then, for $j = 1, \ldots, m$, compute

$$p(z_n = (k^{-n} + j, 1) | \sigma^2_{k^{-n}+j}, \beta_{0,k^{-n}+j}, \lambda_{k^{-n}+j}, \alpha_\theta, \alpha_{\psi,k^{-n}+j}, \tilde{y}_n, x_n)$$
$$\propto \frac{\alpha_\theta}{m} h(\tilde{y}_n | \sigma^2_{k^{-n}+j}, \beta_{0,k^{-n}+j}, \lambda_{k^{-n}+j}) h(x_n).$$

5. Update the allocation variable z_n using the allocation probabilities. All empty clusters are removed, and if one of the m new clusters is selected, set $z_n = (k^{-n} + 1, 1)$ and the parameters $(\sigma^2_{k^{-n}+1}, \beta_{0,k^{-n}+1}, \lambda_{k^{-n}+1}, \alpha_{\psi,k^{-n}+1})$ equal to the parameters of the selected new cluster.

After the full Gibbs sweep for the N allocation variables and after updating the cluster parameters, a Metropolis-Hastings step is performed to improve mixing, by proposing to move an x-cluster to be nested within a different or new y-cluster (as described in Wade et al. [2014]). This step is separated into three possible moves: 1) an x-cluster, among those within y-clusters with more than one x-cluster, is moved to a different y-cluster; 2) an x-cluster, among those within y-clusters with more than one x-cluster, is moved to a new y-cluster; 3) an x-cluster, among those within y-clusters with only one x-cluster, is moved to a different y-cluster. Define

$$k_{x,2+} = \sum_{j=1}^{k} k_j \mathbf{1}(k_j > 1) \quad \text{and} \quad k_{x,1} = \sum_{j=1}^{k} \mathbf{1}(k_j = 1).$$

At every iteration, Move 1 is performed if $k_{x,2+} > 0$. Next, with probability 1/2, Move 2 is performed, otherwise, Move 3 is performed (with the exception that when $k_{x,1} = 0$, Move 2 is performed with probability 1, or when $k_{x,2+} = 0$, Move 3 is performed with probability 1).

1. **Move 1:** an x-cluster (nested within a y-cluster with more than one x-cluster) is uniformly selected with probability $k_{x,2+}^{-1}$ and moved to be nested within a different y-cluster selected uniformly with probability $(k-1)^{-1}$. Let $z_{1:N}^*$ denote the proposed allocations defined by moving x-cluster l in y-cluster j to be nested within y-cluster h for $h \in \{1, \ldots, j-1, j+1, \ldots, k\}$. The acceptance probability is $\min(1, p)$, where

$$p = \frac{\Gamma(N_j - N_{l|j})\Gamma(N_h + N_{l|j})}{\Gamma(N_j)\Gamma(N_h)} \frac{\Gamma(\alpha_{\psi,j} + N_j)\Gamma(\alpha_{\psi,h} + N_h)}{\Gamma(\alpha_{\psi,j} + N_j - N_{l|j})\Gamma(\alpha_{\psi,h} + N_h + N_{l|j})}$$
$$* \frac{\alpha_{\psi,h}}{\alpha_{\psi,j}} \frac{h(\tilde{\boldsymbol{Y}}_j^*|\sigma_j^2, \beta_{0,j}, \lambda_j) h(\tilde{\boldsymbol{Y}}_h^*|\sigma_h^2, \beta_{0,h}, \lambda_h)}{h(\tilde{\boldsymbol{Y}}_j|\sigma_j^2, \beta_{0,j}, \lambda_j) h(\tilde{\boldsymbol{Y}}_h|\sigma_h^2, \beta_{0,h}, \lambda_h)} \frac{k_{x,2+}}{k_{x,2+}^*},$$

   where $\tilde{\boldsymbol{Y}}_j^*$ contains the $N_j - N_{l|j}$ outputs under the proposed cluster allocation $z_{1:N}^*$, with the $N_{l|j}$ points removed from y-cluster j, and similarly, $\tilde{\boldsymbol{Y}}_h^*$ contains the $N_h + N_{l|j}$ outputs under the proposed cluster allocation $z_{1:N}^*$, with the $N_{l|j}$ points added to y-cluster h. The notation $k_{x,2+}^*$ represents the number of x-clusters within a y-cluster with more than one x-cluster under the proposed partition, i.e. $k_{x,2+}^* = k_{x,2+} - \mathbf{1}(k_j = 2) + \mathbf{1}(k_h = 1)$.

2. **Move 2**: an x-cluster (nested within a y-cluster with more than one x-cluster) is uniformly selected with probability $k_{x,2+}^{-1}$ and moved to be nested within a new y-cluster. In this case, we propose new parameters $(\sigma_{k+1}, \beta_{0,k+1}, \lambda_{k+1}, \alpha_{\psi,k+1})$ for the new y-cluster from the prior. The acceptance probability is $\min(1, p)$, where

$$p = \frac{\Gamma(N_j - N_{l|j})\Gamma(N_{l|j})}{\Gamma(N_j)} \frac{\Gamma(\alpha_{\psi,j} + N_j)\Gamma(\alpha_{\psi,k+1})}{\Gamma(\alpha_{\psi,j} + N_j - N_{l|j})\Gamma(\alpha_{\psi,k+1} + N_{l|j})}$$
$$* \alpha_\theta \frac{\alpha_{\psi,k+1}}{\alpha_{\psi,j}} \frac{h(\tilde{\boldsymbol{Y}}_j^*|\sigma_j^2, \beta_{0,j}, \lambda_j) h(\tilde{\boldsymbol{Y}}_{k+1}^*|\sigma_{k+1}^2, \beta_{0,k+1}, \lambda_{k+1})}{h(\tilde{\boldsymbol{Y}}_j|\sigma_j^2, \beta_{0,j}, \lambda_j)} \frac{k_{x,2+}}{k_{x,1}^* k},$$

   where $k_{x,1}^* = k_{x,1} + 1 + \mathbf{1}(k_j = 2)$ represents the number of x-clusters within a y-cluster with only one x-cluster under the proposed partition.

3. **Move 3**: an x-cluster (nested within a y-cluster with only one x-cluster) is uniformly selected with probability $k_{x,1}^{-1}$ and moved to be nested within a different y-cluster selected uniformly with probability $(k-1)^{-1}$. Let $z_{1:N}^*$ denote the proposed allocations defined by moving x-cluster l in y-cluster j to be nested within y-cluster h for $h \in \{1, \ldots, j-1, j+1, \ldots, k\}$. The acceptance

probability is $\min(1, p)$, where

$$p = \frac{\Gamma(N_h + N_j)}{\Gamma(N_h)\Gamma(N_j)} \frac{\Gamma(\alpha_{\psi,j} + N_j)\Gamma(\alpha_{\psi,h} + N_h)}{\Gamma(\alpha_{\psi,h} + N_h + N_j)\Gamma(\alpha_{\psi,j})} \frac{1}{\alpha_\theta} \frac{\alpha_{\psi,h}}{\alpha_{\psi,j}}$$
$$* \frac{h(\tilde{\boldsymbol{Y}}_h^* | \sigma_h^2, \beta_{0,h}, \lambda_h)}{h(\tilde{\boldsymbol{Y}}_j | \sigma_j^2, \beta_{0,j}, \lambda_j) h(\tilde{\boldsymbol{Y}}_h | \sigma_h^2, \beta_{0,h}, \lambda_h)} \frac{k_{x,1}(k-1)}{k_{x,2+}^*}.$$

Mass parameters. The additional mass parameters $\alpha_{\psi,1:k}$ are updated using the auxiliary variable technique of Escobar and West [1995]. Specifically, for $j = 1, \ldots, k$, we sample an auxiliary variable $\xi_j \sim \text{Beta}(\alpha_{\psi,j} + 1, N_j)$,; set

$$\widehat{v}_{\psi,j} = v_\psi - \log(\xi_j) \quad \text{and} \quad \widehat{u}_{\psi,j} = \begin{cases} u_\psi + k_j - 1 & \text{w/ prob } \frac{N_j \widehat{v}_{\psi,j}}{N_j \widehat{v}_{\psi,j} + u_\psi + k_j - 1} \\ u_\psi + k_j & \text{w/ prob } \frac{u_\psi + k_j - 1}{N_j \widehat{v}_{\psi,j} + u_\psi + k_j - 1} \end{cases};$$

and sample $\alpha_{\psi,j} \sim \text{Ga}(\widehat{u}_{\psi,j}, \widehat{v}_{\psi,j})$.

C.6 Predictions for the enriched mixture of generalised GP experts

In the Gaussian example, the posterior density for a new output y_* given a new x_* is again given by

$$f(y_*|x_*, y_{1:N}, x_{1:N}) = \int f(y_*|x_*, y_{1:N}, x_{1:N}, \zeta) \frac{\pi(\zeta|y_{1:N}, x_{1:N}) f(x_*|x_{1:N}, \zeta)}{f(x_*|x_{1:N})} d\zeta$$
$$\approx C^{-1} \left(\sum_{m=1}^{M} p^{(m)}_{k^{(m)}+1}(x_*) h(y_*) + \sum_{j=1}^{k^{(m)}} p^{(m)}_j(x_*) h(y_*|\boldsymbol{Y}^{(m)}_j, \beta^{(m)}_{0,j}, \lambda^{(m)}_j, \sigma^{2\,(m)}_j) \right).$$

but with the more flexible in (5.5). Note again that the marginal likelihood $h(y_*)$ for a new cluster is unavailable in closed form and must be approximated.

Again, through the joint modelling approach, we can compute predictions of y_* based only on a subset of inputs, say only based on a single input x_{*p}, by marginalising over the other inputs. In this case, the weights would only involve the local predictive marginal likelihood of x_{*p} for each cluster $h(x_{*p}|\mathbf{X}^{(m)}_{l|j,p})$, $j = 1, \ldots, k^{(m)}$ and $l = 1, \ldots, k^{(m)}_j$, and for a new cluster $h(x_{*p})$. However, the local expectation would need to be integrated with respect to predictive marginal likelihood of $x_{*-p} = (x_{*1}, \ldots, x_{*p-1}, x_{*p+1}, \ldots, x_{*P})$ in each nested clustering (j, l), with respect to the predictive marginal likelihoods $h(x_{*-p}|\mathbf{X}^{(m)}_{l|j,-p})$ for $j = 1, \ldots, k^{(m)}$ and $l = 1, \ldots, k^{(m)}_j$. For example, in the Gaussian case,

$$\mathbb{E}[y_*|x_{*p}, y_{1:N}, x_{1:N}] \approx C_p^{-1} \left(\sum_{m=1}^{M} p^{(m)}_{k^{(m)}+1}(x_{*p}) \mu_\beta + \sum_{j=1}^{k^{(m)}} p^{(m)}_{j,1}(x_{*p}) \mathbb{E}_{x_{*-p}}[\widehat{m}^{(m)}_j(x_*)] \right.$$
$$\left. + \sum_{j=1}^{k^{(m)}} \sum_{l=1}^{k^{(m)}_j} p^{(m)}_{j,l}(x_{*p}) \mathbb{E}_{x_{*-p}}[\widehat{m}^{(m)}_j(x_*)|\mathbf{X}^{(m)}_{l|j,p}] \right),$$

where expectations are taken with respect to $h(x_{*-p})$ and $h(x_{*-p}|\mathbf{X}^{(m)}_{l|j,-p})$, i.e.

$$\mathbb{E}_{x_{*-p}}[\widehat{m}^{(m)}_j(x_*)] = \int \widehat{m}^{(m)}_j(x_*) \prod_{p' \neq p} h(x_{*p'}) dx_{*-p},$$
$$\mathbb{E}_{x_{*-p}}[\widehat{m}^{(m)}_j(x_*)|\mathbf{X}^{(m)}_{l|j,-p}] = \int \widehat{m}^{(m)}_j(x_*) \prod_{p' \neq p} h(x_{*p'}|\mathbf{X}^{(m)}_{l|j,p'}) dx_{*-p},$$

with

$$p^{(m)}_{k^{(m)}+1}(x_{*p}) = \frac{\alpha^{(m)}_{\theta}}{\alpha^{(m)}_{\theta}+N} h(x_{*p}), \quad p^{(m)}_{j,1}(x_{*p}) = \frac{N^{(m)}_{j}}{\alpha^{(m)}_{\theta}+N} \frac{\alpha^{(m)}_{\psi,j}}{\alpha^{(m)}_{\psi,j}+N^{(m)}_{j}} h(x_{*p})$$

$$p^{(m)}_{j,l}(x_{*p}) = \frac{N^{(m)}_{j}}{\alpha^{(m)}_{\theta}+N} \frac{N^{(m)}_{l|j}}{\alpha^{(m)}_{\psi,j}+N^{(m)}_{j}} h(x_{*p}|\boldsymbol{X}^{(m)}_{l|j,p}),$$

with $C_p = \sum_{m=1}^{M} p^{(m)}_{k^{(m)}+1}(x_{*p}) + \sum_{j=1}^{k^{(m)}} p^{(m)}_{j,1}(x_{*p}) + \sum_{j=1}^{k^{(m)}} \sum_{l=1}^{k^{(m)}_j} p^{(m)}_{j,l}(x_{*p})$.

Bibliography

Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim. On the surprising behavior of distance metrics in high dimensional space. In International conference on database theory, pages 420–434. Springer, 2001.

Amr Ahmed, Moahmed Aly, Joseph Gonzalez, Shravan Narayanamurthy, and Alexander J Smola. Scalable inference in latent variable models. In Proceedings of the fifth ACM international conference on Web search and data mining, pages 123–132. ACM, 2012.

A Al-Tabbaa, J.M Ayotamuno, and R.J Martin. One-dimensional solute transport in stratified sands at short travel distances. Journal of Hazardous Materials, 73 (1):1–15, 2000.

David J Aldous. Exchangeability and related topics. In École d'Été de Probabilités de Saint-Flour XIII1983, pages 1–198. Springer, 1985.

Alaa H. Aly and Richard C. Peralta. Optimal design of aquifer cleanup systems under uncertainty using a neural network and a genetic algorithm. Water Resources Research, 35(8):2523–2532, 1999.

Christophe Andrieu and Gareth O Roberts. The pseudo-marginal approach for efficient Monte Carlo computations. The Annals of Statistics, pages 697–725, 2009.

Christophe Andrieu, Nando De Freitas, Arnaud Doucet, and Michael I Jordan. An introduction to mcmc for machine learning. Machine learning, 50(1-2):5–43, 2003.

Behzad Ataie-Ashtiani, Hamed Ketabchi, and Mohammad Mahdi Rajabi. Optimal management of a freshwater lens in a small island using surrogate models and evolutionary algorithms. Journal of Hydrologic Engineering, 19(2):339–354, 2014.

Ivo Babuška, Fabio Nobile, and Raul Tempone. A stochastic collocation method for elliptic partial differential equations with random input data. SIAM Journal on Numerical Analysis, 45(3):1005–1034, 2007.

William Barcella, Maria De Iorio, and Gianluca Baio. A comparative review of variable selection techniques for covariate dependent Dirichlet process mixture models. Canadian Journal of Statistics, 45:254–273, 2017.

D.A. Barry, H. Prommer, C.T. Miller, P. Engesgaard, A. Brun, and C. Zheng. Modelling the fate of oxidisable organic contaminants in groundwater. Advances in Water Resources, 25(8):945–983, 2002.

Domenico A. Bau and Alex S. Mayer. Stochastic management of pump-and-treat strategies using surrogate functions. Advances in Water Resources, 29(12):1901–1917, 2006.

Matthias Bauer, Mark van der Wilk, and Carl Edward Rasmussen. Understanding probabilistic sparse gaussian process approximations. In Advances in neural information processing systems, pages 1533–1541, 2016.

Mark A Beaumont. Estimation of population growth or decline in genetically monitored populations. Genetics, 164(3):1139–1160, 2003.

Michael Betancourt. A conceptual introduction to hamiltonian monte carlo. arXiv preprint arXiv:1701.02434, 2017.

Michael Betancourt, Simon Byrne, Sam Livingstone, Mark Girolami, et al. The geometric foundations of hamiltonian monte carlo. Bernoulli, 23(4A):2257–2298, 2017.

Wolfgang Betz, Iason Papaioannou, and Daniel Straub. Numerical methods for the discretization of random fields by means of the Karhunen-Loeve expansion. Computer Methods in Applied Mechanics and Engineering, 271:109–129, 2014.

Rajib Kumar Bhattacharjya and Bithin Datta. Optimal management of coastal aquifers using linked simulation optimization approach. Water Resources Management, 19(3):295–320, 2005.

Christopher M Bishop. Variational principal components. 1999.

Sebastian Bitzer and Christopher KI Williams. Kick-starting GPLVM optimization via a connection to metric MDS. In NIPS 2010 Workshop on Challenges of Data Visualization, 2010.

David Blackwell and James B MacQueen. Ferguson distributions via pólya urn schemes. The annals of statistics, pages 353–355, 1973.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. Journal of the American Statistical Association, 112, 2017.

D. Boi, V. Stankovi, M. Gorgievski, G. Bogdanovi, and R. Kovaevi. Adsorption of heavy metal ions by sawdust of deciduous trees. Journal of Hazardous Materials, 171(1):684–692, 2009.

E. Borgonovo, W. Castaings, and S. Tarantola. Model emulation and moment-independent sensitivity analysis: An application to environmental modelling. Environmental Modelling & Software, 34:105–115, 2012.

Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng. Handbook of markov chain monte carlo. CRC press, 2011.

Michael A Celia, Lajpat R Ahuja, and George F Pinder. Orthogonal collocation and alternating-direction procedures for unsaturated flow problems. Advances in Water Resources, 10(4):178–187, 1987.

Michael A. Celia, Efthimios T. Bouloutas, and Rebecca L. Zarba. A general mass-conservative numerical solution for the unsaturated flow equation. Water Resources Research, 26(7):1483–1496, 1990.

A.B. Chan and D. Dong. Generalized gaussian process models. In IEEE Conf. Computer Vision and Pattern Recognition, pages 2681–2688, 2011.

Siddhartha Chib and Edward Greenberg. Understanding the metropolis-hastings algorithm. The american statistician, 49(4):327–335, 1995.

G. Consonni and P. Veronese. Conditionally reducible natural exponential families and enriched conjugate priors. Scandinavian Journal of Statistics, 28:377–406, 2001.

Stefano Conti and Anthony O'Hagan. Bayesian emulation of complex multi-output and dynamic computer models. Journal of Statistical Planning and Inference, 140 (3):640–651, 2010.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. Machine learning, 20(3):273–297, 1995.

Lehel Csató and Manfred Opper. Sparse on-line gaussian processes. Neural computation, 14(3):641–668, 2002.

C Currin, T Mitchell, M Morris, and D Ylvisaker. A Bayesian approach to the design and analysis of computer experiments. Technical report, ORNL-6498, Oak Ridge National Laboratory, 1988.

R. Daley. Atmospheric data analysis. Cambridge University Press, Cambridge, UK, 1991.

Andreas Damianou. Deep Gaussian processes and variational propagation of uncertainty. PhD thesis, University of Sheffield, 2015.

Andreas Damianou and Neil Lawrence. Deep Gaussian processes. In Artificial Intelligence and Statistics, pages 207–215, 2013.

Andreas Damianou, Michalis K Titsias, and Neil D Lawrence. Variational Gaussian process dynamical systems. In Advances in Neural Information Processing Systems, pages 2510–2518, 2011.

Alexander Graeme de Garis Matthews. Scalable Gaussian process inference using variational methods. PhD thesis, Ph. D. thesis, Department of Engineering, University of Cambridge, 2016.

Christopher C Drovandi, Matthew T Moores, and Richard J Boys. Accelerating pseudo-marginal mcmc using gaussian processes. Computational Statistics & Data Analysis, 118:1–17, 2018.

Simon Duane, Anthony D Kennedy, Brian J Pendleton, and Duncan Roweth. Hybrid monte carlo. Physics letters B, 195(2):216–222, 1987.

David Duvenaud. Automatic model construction with Gaussian processes. PhD thesis, University of Cambridge, 2014.

M.D. Escobar and M. West. Bayesian density estimation and inference using mixtures. Journal of the American Statistical Association, 90:577–588, 1995.

Thomas S Ferguson. A bayesian analysis of some nonparametric problems. The annals of statistics, pages 209–230, 1973a.

T.S. Ferguson. A Bayesian analysis of some nonparametric problems. Annals of Statistics, 1:209–230, 1973b.

Jan Feyen, Diederik Jacques, Anthony Timmerman, and Jan Vanderborght. Modelling water flow and solute transport in heterogeneous soils: A review of recent approaches. Journal of Agricultural Engineering Research, 70(3):231–256, 1998.

Maurizio Filippone. Bayesian inference for gaussian process classifiers with annealing and exact-approximate mcmc. arXiv preprint arXiv:1311.7320, 2013.

Maurizio Filippone and Mark Girolami. Pseudo-marginal Bayesian inference for Gaussian processes. IEEE transactions on pattern analysis and machine intelligence, 36(11):2214–2226, 2014.

K.Y. Foo and B.H. Hameed. An overview of landfill leachate treatment via activated carbon adsorption process. Journal of Hazardous Materials, 171(1):54–60, 2009.

Jianlin Fu and J. Jaime Gomez-Hernandez. Uncertainty assessment and data worth in groundwater flow and mass transport modeling using a blocking markov chain monte carlo method. Journal of Hydrology, 364(3):328–341, 2009.

C. Gadd, S. Wade, A. Shah, and D. Grammatopoulos. Pseudo-marginal Bayesian inference for supervised Gaussian process latent variable models. ArXiv e-prints, March 2018.

C. Gadd, W. Xing, M. Mousavi Nezhad, and A. A. Shah. A surrogate modelling approach based on nonlinear dimension reduction for uncertainty quantification in groundwater flow models. Transport in Porous Media, May 2018. ISSN 1573-1634. doi: 10.1007/s11242-018-1065-7. URL `https://doi.org/10.1007/s11242-018-1065-7`.

Lynn W. Gelhar. Stochastic subsurface hydrology from theory to applications. Water Resources Research, 22(9S):135S–145S, 1986.

Lynn W. Gelhar and Carl L. Axness. Three-dimensional stochastic analysis of macrodispersion in aquifers. Water Resources Research, 19(1):161–180, 1983.

Andrew Gelman. Bayesian model-building by pure thought: some principles and examples. Statistica Sinica, pages 215–232, 1996.

Andrew Gelman, John B Carlin, Hal S Stern, and Donald B Rubin. Bayesian data analysis. Chapman and Hall/CRC, 1995.

Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. IEEE Transactions on pattern analysis and machine intelligence, (6):721–741, 1984.

Roger G Ghanem and Pol D Spanos. Stochastic Finite Elements: A Spectral Approach. Springer, New York, 2003.

Agathe Girard and Roderick Murray-Smith. Gaussian processes: prediction at a noisy input and application to iterative multiple-step ahead forecasting of time-series. Switching and Learning in Feedback Systems, Lecture Notes in Computer Science, Springer, pages 546–551, 2003.

Robert B Gramacy and Herbert K H Lee. Bayesian treed gaussian process models with an application to computer modeling. Journal of the American Statistical Association, 103(483):1119–1130, 2008.

Peter J Green. Reversible jump markov chain monte carlo computation and bayesian model determination. Biometrika, 82(4):711–732, 1995.

Heikki Haario, Eero Saksman, Johanna Tamminen, et al. An adaptive metropolis algorithm. Bernoulli, 7(2):223–242, 2001.

L.A. Hannah, D.M. Blei, and W.B. Powell. Dirichlet process mixtures of generalized linear models. Journal of Machine Learning Research, 12:1923–1953, 2011.

W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. 1970.

Roland Haverkamp, Michel Vauclin, Jaoudat Touma, PJ Wierenga, and Georges Vachaud. A comparison of numerical simulation models for one-dimensional infiltration. Soil Science Society of America Journal, 41(2):285–294, 1977.

Thomas Hemker, Kathleen R. Fowler, Matthew W. Farthing, and Oskar von Stryk. A mixed-integer simulation-based optimization approach with surrogate functions in water resources management. Optimization and Engineering, 9(4):341–360, 2008.

James Hensman, Alexander G Matthews, Maurizio Filippone, and Zoubin Ghahramani. Mcmc for variationally sparse gaussian processes. In Advances in Neural Information Processing Systems, pages 1648–1656, 2015.

Daan Herckenrath, Christian D. Langevin, and John Doherty. Predictive uncertainty analysis of a saltwater intrusion model using null-space monte carlo. Water Resources Research, 47(5), 2011. W05504.

D. Higdon, J. Gattiker, B. Williams, and M. Rightley. Computer model calibration using high-dimensional output. Journal of the American Statistical Association, 103(482):570–583, 2008.

K Huang, BP Mohanty, and M Th Van Genuchten. A new convergence criterion for the modified picard iteration method to solve the variably saturated flow equation. Journal of Hydrology, 178(1-4):69–91, 1996.

J. Hwang, X. Shen, and Y. Pawitan. Mini-mental state examination change score prediction for early diagnosis of Alzheimer's disease. 2014. URL `https://www.synapse.org/#!Synapse:syn2759392/wiki/69612`.

Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. Neural computation, 3(1):79–87, 1991.

S. Jain and R.M. Neal. A split-merge Markov chain Monte Carlo procedure for the Dirichlet process mixture model. Journal of Computational and Graphical Statistics, 13:158–182, 2004.

S. Jain and R.M. Neal. Splitting and merging components of a nonconjugate Dirichlet process mixture model. Bayesian Analysis, 2:445–472, 2007.

Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. Machine learning, 37(2):183–233, 1999.

George P. Karatzas. Developments on modeling of groundwater flow and contaminant transport. Water Resources Management, 31(10):3235–3244, 2017.

Hamed Ketabchi and Behzad Ataie-Ashtiani. Review: Coastal groundwater optimization—advances, challenges, and practical solutions. Hydrogeology Journal, 23(6):1129–1154, 2015.

J.H. Kotecha and P.M. Djuric. Gibbs sampling approach for generation of truncated multivariate gaussian random variables. In 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing, volume 3, pages 1757–1760, 1999.

George Kourakos and Thomas Harter. Parallel simulation of groundwater non-point source pollution using algebraic multigrid preconditioners. Computational Geosciences, 18(5):851–867, 2014.

George Kourakos and Aristotelis Mantoglou. Pumping optimization of coastal aquifers based on evolutionary algorithms and surrogate modular neural network models. Advances in Water Resources, 32(4):507–521, 2009.

George Kourakos, Frank Klein, Andrea Cortis, and Thomas Harter. A groundwater nonpoint source pollution modeling framework to evaluate long-term dynamics of pollutant exceedance probabilities in wells and other discharge locations. Water Resources Research, 48(6):W00L13, 2012.

Andreas H. Kristensen, Tjalfe G. Poulsen, Lars Mortensen, and Per Moldrup. Variability of soil potential for biodegradation of petroleum hydrocarbons in a heterogeneous subsurface. Journal of Hazardous Materials, 179(1):573–580, 2010.

Harold J Kushner and G George Yin. Stochastic approximation algorithms and applications, volume 35 of applications of mathematics, 1997.

Neil Lawrence. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. The Journal of Machine Learning Research, 6: 1783–1816, 2005.

Neil D Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. In Advances in neural information processing systems, pages 329–336, 2004.

Hongyu Li, Li Teng, Wenbin Chen, and I-Fan Shen. Supervised learning on local tangent space. Advances in Neural Networks–ISNN 2005, Lecture Notes in Computer Science, Springer, pages 546–551, 2005.

Fredrik Lindsten and Arnaud Doucet. Pseudo-marginal hamiltonian monte carlo. arXiv preprint arXiv:1607.02516, 2016.

Xiaoming Liu, Jianwei Yin, Zhilin Feng, and Jinxiang Dong. Incremental manifold learning via tangent space alignment. In Friedhelm Schwenker and Simone Marinai, editors, Artificial Neural Networks in Pattern Recognition, pages 107–121, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

Albert Y Lo. On a class of bayesian nonparametric estimates: I. density estimates. The annals of statistics, pages 351–357, 1984.

Xiang Ma and Nicholas Zabaras. An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations. Journal of Computational Physics, 228(8):3084–3113, 2009.

Xiang Ma and Nicholas Zabaras. Kernel principal component analysis for stochastic input model generation. Journal of Computational Physics, 230(19):7311–7331, 2011.

David JC MacKay. Hyperparameters: optimise or integrate out. Maximum entropy and Bayesian methods, 1994.

G. Malsiner-Walli, S. Frühwirth-Schnatter, and B. Grün. Model-based clustering based on sparse finite Gaussian mixtures. Statistics and Computing, 26:303–324, 2016.

Georges Matheron. The intrinsic random functions and their applications. Advances in Applied Probability, pages 439–468, 1973.

Reed M. Maxwell, Claire Welty, and Ronald W. Harvey. Revisiting the cape cod bacteria injection experiment using a stochastic modeling approach. Environmental Science & Technology, 41(15):5548–5558, 2007.

P. McCullagh and J.A. Nelder. Generalized Linear Models. London, 1989.

Edward Meeds and Simon Osindero. An alternative infinite mixture of gaussian process experts. Advances in Neural Information Processing Systems, 18:883, 2006.

Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. The journal of chemical physics, 21(6):1087–1092, 1953.

D. Meyer, E. Dimitriadou, K. Hornik, A. Weingessel, and F. Leisch. e1071: Misc functions of the Department of Statistics, Probability Theory Group (Formerly: E10 2018. URL `https://CRAN.R-project.org/package=e1071`.

Iain Murray and Matthew Graham. Pseudo-marginal slice sampling. In Artificial Intelligence and Statistics, pages 911–919, 2016.

Iain Murray, Ryan Prescott Adams, and David JC MacKay. Elliptical slice sampling. 2010.

M. Myllymäki, A. Särkkä, and A. Vehtari. Hierarchical second-order analysis of replicated spatial point patterns with non-spatial covariates. Spatial Statistics, 8: 104–121, 2014.

Andrew Naish-Guzman and Sean Holden. The generalized fitc approximation. In Advances in Neural Information Processing Systems, pages 1057–1064, 2008.

Radford M Neal. Markov chain sampling methods for dirichlet process mixture models. Journal of computational and graphical statistics, 9(2):249–265, 2000.

Radford M Neal et al. Mcmc using hamiltonian dynamics. Handbook of Markov Chain Monte Carlo, 2(11):2, 2011.

Trung Nguyen and Edwin Bonilla. Fast allocation of gaussian process experts. In Proceedings of The 31st International Conference on Machine Learning, pages 145–153, 2014.

Hannes Nickisch and Carl Edward Rasmussen. Approximations for binary gaussian process classification. Journal of Machine Learning Research, 9(Oct):2035–2078, 2008.

Fabio Nobile, Raúl Tempone, and Clayton G Webster. A sparse grid stochastic collocation method for partial differential equations with random input data. SIAM Journal on Numerical Analysis, 46(5):2309–2345, 2008.

Anthony O'Hagan and JFC Kingman. Curve fitting and optimal design for prediction. Journal of the Royal Statistical Society. Series B (Methodological), pages 1–42, 1978.

Evan K. Paleologos, T. Avanidou, and N. Mylopoulos. Stochastic analysis and prioritization of the influence of parameter uncertainty on the predicted pressure profile in heterogeneous, unsaturated soils. Journal of Hazardous Materials, 136 (1):137–143, 2006.

Michail Papathomas, John Molitor, Clive Hoggart, David Hastie, and Sylvia Richardson. Exploring data from genetic association studies using Bayesian variable selection and the Dirichlet process: application to searching for gene× gene patterns. Genetic epidemiology, 36:663–674, 2012.

Jim Pitman. Some developments of the blackwell-macqueen urn scheme. Lecture Notes-Monograph Series, pages 245–267, 1996.

Christopher A Pope, John Paul Gosling, Stuart Barber, Jill Johnson, Takanobu Yamaguchi, Graham Feingold, and Paul Blackwell. Modelling spatial heterogeneity and discontinuities using voronoi tessellations. arXiv preprint arXiv:1802.05530, 2018.

Joaquin Quiñonero-Candela and Carl Edward Rasmussen. A unifying view of sparse approximate gaussian process regression. Journal of Machine Learning Research, 6(Dec):1939–1959, 2005.

Mohammad Mahdi Rajabi, Behzad Ataie-Ashtiani, and Craig T. Simmons. Polynomial chaos expansions for uncertainty propagation and moment independent sensitivity analysis of seawater intrusion simulations. Journal of Hydrology, 520: 101–122, 2015.

Carl Edward Rasmussen. Gaussian processes in machine learning. In Advanced lectures on machine learning, pages 63–71. Springer, 2004.

Carl Edward Rasmussen and Zoubin Ghahramani. Infinite mixtures of gaussian process experts. Advances in neural information processing systems, 2:881–888, 2002.

C.E. Rasmussen and C.K.I. Williams. Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning). The MIT Press, 2005.

Klaus Rathfelder and Linda M. Abriola. Mass conservative numerical solutions of the head-based Richards equation. Water Resources Research, 30(9):2579–2586, 1994.

C Ray and BP Mohanty. Some numerical investigations of the Richards' equation. ASAE Paper 92-2586, American Society of Agricultural Engineers, St Joseph, MI, 1992.

Yordan P Raykov, Alexis Boukouvalas, and Max A Little. Simple approximate map inference for dirichlet processes. arXiv preprint arXiv:1411.0939, 2014.

Saman Razavi, Bryan A. Tolson, and Donald H. Burn. Review of surrogate modeling in water resources. Water Resources Research, 48(7), 2012. ISSN 1944-7973. W07401.

Herbert Robbins and S Monro. [a]a stochastic approximation method, [o] annals math. Statistics, 22:400–407, 1951.

Christian P Robert and George Casella. The metropolishastings algorithm. In Monte Carlo Statistical Methods, pages 231–283. Springer, 1999.

Jerome Sacks, William J Welch, Toby J Mitchell, and Henry P Wynn. Design and analysis of computer experiments. Statistical Science, 4(4):409–423, 1989.

Hugh Salimbeni and Marc Deisenroth. Doubly stochastic variational inference for deep gaussian processes. In Advances in Neural Information Processing Systems, pages 4588–4599, 2017.

TJ Santner, BJ Williams, and WI Notz. The design and analysis of computer experiments springer-verlag. New York. 283pp, 2003.

Dirk Schfer, Wolfgang Schfer, and Wolfgang Kinzelbach. Simulation of reactive processes related to biodegradation in aquifers. Journal of Contaminant Hydrology, 31(1):167–186, 1998.

Matthias Seeger, Neil D Lawrence, and Ralf Herbrich. Fast sparse gaussian process methods: The informative vector machine. In Advances in neural information processing systems, pages 625–632, 2003.

J. Sethuraman. A constructive definition of dirichlet priors. Statistica Sinica, 4: 639–650, 1994a.

Jayaram Sethuraman. A constructive definition of dirichlet priors. Statistica sinica, pages 639–650, 1994b.

Burr Settles. Active learning. Synthesis Lectures on Artificial Intelligence and Machine Learning, 6(1):1–114, 2012.

Hamid Taheri Shahraiyni and Behzad Ataie-Ashtiani. Mathematical forms and numerical schemes for the solution of unsaturated flow equations. Journal of Irrigation and Drainage Engineering, 138(1):63–72, 2011.

SheffieldML. vargplvm. `https://github.com/SheffieldML/vargplvm`, 2017.

Jeffrey S Simonoff. Smoothing Methods in Statistics. Springer, New York, 1996.

Sergey A Smolyak. Quadrature and interpolation formulas for tensor products of certain classes of functions. In Dokl. Akad. Nauk SSSR, volume 4, pages 240–243, 1963.

Edward Snelson and Zoubin Ghahramani. Sparse gaussian processes using pseudo-inputs. In Advances in neural information processing systems, pages 1257–1264, 2006a.

Edward Snelson and Zoubin Ghahramani. Variable noise and dimensionality reduction for sparse Gaussian processes. In Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence, pages 461–468, 2006b.

J. Sreekanth and Bithin Datta. Comparative evaluation of genetic programming and neural network as potential surrogate models for coastal aquifer management. Water Resources Management, 25(13):3201–3218, 2011a.

J. Sreekanth and Bithin Datta. Coupled simulation-optimization model for coastal aquifer management using genetic programming-based ensemble surrogate models and multiple-realization optimization. Water Resources Research, 47(4), 2011b. W04516.

J. Sreekanth and Bithin Datta. Stochastic and robust multi-objective optimal management of pumping from coastal aquifers under parameter uncertainty. Water Resources Management, 28(7):2005–2019, 2014.

Michalis K Titsias. Variational learning of inducing variables in sparse Gaussian processes. In International Conference on Artificial Intelligence and Statistics, pages 567–574, 2009.

Michalis K Titsias and Neil D Lawrence. Bayesian Gaussian process latent variable model. In International Conference on Artificial Intelligence and Statistics, pages 844–851, 2010.

Michalis K Titsias, Neil D Lawrence, and Magnus Rattray. Efficient sampling for gaussian process inference using control variables. In Advances in Neural Information Processing Systems, pages 1681–1688, 2009.

Volker Tresp. Mixtures of gaussian processes. In Advances in neural information processing systems, pages 654–660, 2001.

Richard E Turner and Maneesh Sahani. Two problems with variational expectation maximisation for time-series models. Bayesian Time series models, 1(3.1):3–1, 2011.

Aki Vehtari, Andrew Gelman, and Jonah Gabry. Pareto smoothed importance sampling. arXiv preprint arXiv:1507.02646, 2015.

W. N. Venables and B. D. Ripley. Modern Applied Statistics with S. Springer, New York, fourth edition, 2002.

Efstratios G. Vomvoris and Lynn W. Gelhar. Stochastic analysis of the concentration variability in a three-dimensional heterogeneous aquifer. Water Resources Research, 26(10):2591–2602, 1990.

S. Wade. mcclust.ext: Point estimation and credible balls for Bayesian cluster analysis, 2015. URL https://www.researchgate.net/publication/279848500_mcclustext-manual.

S. Wade and Z. Ghahramani. Bayesian cluster analysis: point estimation and credible balls. Bayesian Analysis, 2017.

S. Wade, S. Mongelluzzo, and S. Petrone. An enriched conjugate prior for Bayesian nonparametric inference. Bayesian Analysis, 6:359–386, 2011.

Sara Wade, David B Dunson, Sonia Petrone, and Lorenzo Trippa. Improving prediction from dirichlet process mixtures via enrichment. The Journal of Machine Learning Research, 15(1):1041–1071, 2014.

Xiaoliang Wan and George Em Karniadakis. A sharp error estimate for the fast Gauss transform. Journal of Computational Physics, 219(1):7–12, 2006.

Jia Wei, Hong Peng, Yi-Shen Lin, Zhi-Mao Huang, and Jia-Bing Wang. Adaptive neighborhood selection for manifold learning. In Machine Learning and Cybernetics, 2008 International Conference on, volume 1, pages 380–384. IEEE, 2008.

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In Proceedings of the 28th International Conference on Machine Learning (ICML-11), pages 681–688, 2011.

David H Wolpert and William G Macready. No free lunch theorems for optimization. IEEE transactions on evolutionary computation, 1(1):67–82, 1997.

E. Wong. Stochastic Processes in Information and Dynamical Systems. McGraw-Hill, New York, 1971.

Wei Xing, Akeel A Shah, and Prasanth B Nair. Reduced dimensional Gaussian process emulators of parametrized partial differential equations based on Isomap. Proceedings of the Royal Society of London A, 471(2174):20140697, 2015.

WW Xing, V Triantafyllidis, AA Shah, PB Nair, and Nicholas Zabaras. Manifold learning for the emulation of spatial fields from computational models. Journal of Computational Physics, 326:666–690, 2016.

Dongbin Xiu. Efficient collocational approach for parametric uncertainty analysis. Communications in Computational Physics, 2(2):293–309, 2007.

Dongbin Xiu and Jan S Hesthaven. High-order collocation methods for differential equations with random inputs. SIAM Journal on Scientific Computing, 27(3): 1118–1139, 2005.

Dongbin Xiu and George Em Karniadakis. The Wiener–Askey polynomial chaos for stochastic differential equations. SIAM Journal on Scientific Computing, 24(2): 619–644, 2002.

Yaming Yu and Xiao-Li Meng. To center or not to center: That is not the question–an ancillarity-sufficiency interweaving strategy (asis) for boosting mcmc efficiency. Journal of Computational and Graphical Statistics, 20(3):531–570, 2011.

Chao Yuan and Claus Neubauer. Variational mixture of gaussian process experts. In Advances in Neural Information Processing Systems, pages 1897–1904, 2009.

Raaecca L Zarba, ET Bouloutas, and M Celia. General mass-conservative numerical solution for the unsaturated flow equation. Water Resources Research WRERAQ, 26(7):1483–1496, 1990.

Yubin Zhan and Jianping Yin. Robust local tangent space alignment via iterative weighted PCA. Neurocomputing, 74(11):1985–1993, 2011.

Cheng Zhang, Judith Butepage, Hedvig Kjellstrom, and Stephan Mandt. Advances in variational inference. arXiv preprint arXiv:1711.05597, 2017.

Dongxiao Zhang and Zhiming Lu. An efficient, high-order perturbation approach for flow in random porous media via Karhunen-Loeve and polynomial expansions. Journal of Computational Physics, 194(2):773–794, 2004.

Z. Zhang, J. Wang, and H. Zha. Adaptive manifold learning. IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(2):253–265, 2012.

Zhenyue Zhang and Hongyuan Zha. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. SIAM Journal on Scientific Computing, 26(1):313–338, 2004.

F. Zhu and Y. Guan. Guanlab's solution to the 2014 DREAM Alzheimer's disease big data challenge (1st place). 2014. URL `https://www.synapse.org/#!Synapse:syn2527678/wiki/69937`.

Xianlin Zou and Qingsheng Zhu. Adaptive neighborhood graph for LTSA learning algorithm without freeparameter. International Journal of Computer Applications, 19(4):28–33, 2011.